

23-2208

**United States Court of Appeals
for the Federal Circuit**

FINTIV, INC.,

Plaintiff-Appellant

v.

APPLE INC.,

Defendant-Appellees

Appeal from the United States District Court for the Western District of Texas,
Case No. 1:21-cv-00896-ADA, District Judge Alan D. Albright

APPELLANT FINTIV, INC.'S NON-CONFIDENTIAL OPENING BRIEF

Meredith Martin Addy
Charles A. Pannell, III
ADDYHART P.C.
10 Glenlake Parkway, Suite 130
Atlanta, Georgia 30328
312.320.4200
meredith@addyhart.com
cpannell@addyhart.com

Caren A. Yusem
ADDYHART P.C.
1101 Pennsylvania Avenue, N.W.
Suite 300
Washington, DC 20004
312.804.4885
caren@addyhart.com

Jonathan K. Waldrop
Darcy L. Jones
Marcus A. Barber
John W. Downing
Heather S. Kim
ThucMinh Nguyen
Kasowitz Benson Torres LLP
333 Twin Dolphin Drive, Suite 200
Redwood Shores, CA 94065
650.453.5170
jwaldrop@kasowitz.com
djones@kasowitz.com
mbarber@kasowitz.com
jdowning@kasowitz.com
hkim@kasowitz.com
tnguyen@kasowtiz.com

Attorneys for Appellant Fintiv, Inc.

(continued on next page)

November 16, 2023

Paul G. Williams
Kasowitz Benson Torres LLP
1230 Peachtree Street, NE
Suite 2445
Atlanta, GA 30309
404.260.6102
pwilliams@kasowitz.com

Attorneys for Appellant Fintiv, Inc.
(continued from previous page)

November 16, 2023

**EXEMPLARY PATENT CLAIMS 11, 18, & 23
OF U.S. PATENT NO. 8,843,125**

11. A method for provisioning a contactless card applet in a mobile device comprising a mobile wallet application, the method comprising:

- activating the mobile wallet application;
- connecting to a Trusted Service Manager (TSM) system;
- synchronizing the mobile wallet application with the TSM system;
- displaying a contactless card applet based on attributes of the mobile device;
- receiving a selection of a contactless card applet;
- retrieving a widget and a wallet management applet (WMA) corresponding to the contactless card applet; and
- provisioning the selected contactless card applet, the widget, and the WMA.

Appx00098, 13:15-29.

18. A wallet management system (WMS) in a non-transitory storage medium to store and manage mobile wallet account information, comprising:

- a wallet client management component configured to store and to manage a mobile wallet application;
- a widget management component configured to store and to manage widgets;

a device profile management component configured to store mobile device information; and

a rule engine configured to filter a widget based on the mobile device information,

wherein said wallet management system is configured to receive the mobile device information from a mobile device and store the mobile device information in the device profile management component, and

wherein said wallet management system is configured to register the mobile device and the mobile wallet application in a Trusted Service Manager (TSM) system.

Appx00098, 14:7-23.

23. A mobile device, comprising:

a secure element (SE);

a mobile wallet application configured to store a widget corresponding to a contactless card applet, wherein the contactless card applet is stored in the SE;

a wallet management applet (WMA) corresponding to the contactless card applet, wherein the WMA is stored in the SE; and

an over-the-air (OTA) proxy configured to provision the contactless card applet, a widget corresponding to the contactless card applet, and the WMA,

wherein said OTA proxy is configured to capture mobile device information comprising SE information; and

wherein said OTA proxy is configured to transmit the mobile device information for registering the mobile wallet application.

Appx00098, 14:36-53.

CERTIFICATE OF INTEREST

Counsel for Appellant Fintiv, Inc., certifies:

1. **Represented Entities.** Provide the full names of all entities represented by undersigned counsel in this case. Fed. Cir. R. 47.4(a)(1).

Fintiv, Inc.

2. **Real Party in Interest.** Provide the full names of all real parties in interest for the entities. Do not list the real parties if they are the same as the entities. Fed. Cir. R. 47.4(a)(2).

None/Not Applicable.

3. **Parent Corporations and Stockholders.** Provide the full names of all parent corporations for the entities and all publicly held companies that own 10 percent or more of the stock of the entities:

None/Not Applicable.

4. **Legal Representatives.** List all law firms, partners, and associates that (a) appeared for the entities in the originating court or agency or (b) are expected to appear in this court for the entities. Do not include those who have already entered an appearance in this court. Fed. Cir. 4. 47.4(a)(4).

(a) Marc E. Kasowitz, Kasowitz Benson Torres LLP
Jeceaca An, Kasowitz Benson Torres LLP
Julianne Laporte, Kasowitz Benson Torres LLP
Chen Jia, Kasowitz Benson Torres LLP

Jack Shaw, formerly Kasowitz Benson Torres LLP
Daniel C. Miller, formerly Kasowitz Benson Torres LLP
Rodney R. Miller, formerly Kasowitz Benson Torres LLP
Gurtej Singh, formerly Kasowitz Benson Torres LLP
Shelley Ivan, formerly Kasowitz Benson Torres LLP
Trevor J. Welch, formerly Kasowitz Benson Torres LLP

Raymond W. Mort, III, The Mort Law Firm, PLLC

J. Mark Mann, Mann, Tindel & Thompson
Andy W. Tindel, Mann, Tindel & Thompson
G. Blake Thompson, Mann, Tindel & Thompson

Craig D. Cherry, Cherry Johnson Siegmund James PLLC
Justin W. Allen, Cherry Johnson Siegmund James PLLC

George Philip Cowden, The Cosden Law Firm PLLC

(b) Not applicable.

5. **Related Cases.** Provide the case titles and numbers of any case known to be pending in this court or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal. *See* Fed. Cir. R. 47.4(a)(5) and 47.5(b).

None.

6. **Organizational Victims and Bankruptcy Cases.** Provide any information required under Fed. R. App. P. 26.1(b) (organizational victims in criminal cases) and 26.1(c) (bankruptcy case debtors and trustees). *See* Fed. Cir. R. 47.4(a)(6).

None/Not Applicable.

Dated: November 16, 2023

By: /s/ Meredith Martin Addy
Meredith Martin Addy

TABLE OF CONTENTS

EXEMPLARY PATENT CLAIMS 11, 18, & 23 OF U.S. PATENT NO. 8,843,125	i
CERTIFICATE OF INTEREST	iv
TABLE OF CONTENTS	vi
TABLE OF AUTHORITIES	ix
STATEMENT OF RELATED CASES	1
STATEMENT OF JURISDICTION	1
STATEMENT OF THE ISSUES	1
STATEMENT OF THE CASE	2
I. The Invention and the '125 Patent	2
A. The Problems to be Solved	4
B. The '125 Patent Solution.....	5
1. Server-Side Components	6
2. Mobile Device Components	8
3. The Representative Patent Claims	10
II. The Accused Apple Products	10
A. Development of Apple's Wallet and Pay Products and Services	10
B. Overview of Apple's Wallet and Apple Pay Products and Services .	11
III. The Proceedings Below	17
A. Claim Construction	17
B. Initial Summary Judgment Decision.....	18
C. Reconsideration of Summary Judgment Decision	18
SUMMARY OF THE ARGUMENT	22
STANDARD OF REVIEW	23

ARGUMENT.....	24
I. The District Court Erred by Demanding Evidence of “Widget” Source Code	24
A. Direct Evidence of Source Code is not Required to Prove Infringement.....	25
B. The Demand for “Widget” Source Code Evidence Overlooks the Specific Claim Language.....	28
C. The Claim Construction of “Widget” Does Not Require “Widget” Code	32
II. Even with the District Court’s Requirement of Code, Genuine Issues of Fact Confirm the Existence of a “Widget.”.....	34
A. Fintiv and its Expert have Specifically Identified the “Widget” Software Through Tests and Examples from the Accused Products .	36
1. Dr. Shamos Demonstrated how the Functionality of Apple’s Wallet and Apple Pay Used the Claimed “Widget”	36
2. Circumstantial Evidence Indicates that Apple’s Accused Products Use “Widgets”	43
B. The District Court’s Criticisms of Fintiv’s Evidence Failed to Consider the Proper Context, Much Less Overcome an Inference in Favor of Fintiv	46
1. Dr. Shamos’s Statement that he did not Identify a Specific “Widget” Source Code File was Not a Concession that there was No Widget	47
2. The District Court Erred in Concluding that no Expert Personally Reviewed the Apple Source Code to Support Dr. Shamos’s Opinions	50
3. The District Court’s Factual Resolution that Source Code Files with Code ID in the File Name Could not be “Widgets” is Contradicted by the Evidence and Apple’s Own Testimony	52
4. The District Court’s Findings Regarding Other Widget Evidence were Improper and Incorrect.....	54

CONFIDENTIAL MATERIAL OMITTED

5.	The District Court made Improper Findings of Fact About Related “Widget” Evidence	56
III.	Approving the District Court’s Demand for Source Code Sets An Extreme and Unworkable Precedent in Patent Litigation	60
	CONCLUSION	64
	CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATIONS	a
	ADDENDUM	b
	CERTIFICATE OF CONFIDENTIAL MATERIAL	c
	CERTIFICATE OF SERVICE	d

CONFIDENTIAL MATERIAL OMITTED

The confidential information on pages vii, 21, 23, 26, 37, 43, 44, 45, 46, 48, 49, 52, 53, 54, 55, and 59 are words that identify code files or images from Apple technical documentation about its software that Apple has asserted are confidential and covered by the district court protective order.

TABLE OF AUTHORITIES

Cases

<i>Absolute Software, Inc. v. Stealth Signal, Inc.</i> , 659 F.3d 1121 (Fed. Cir. 2011)	23
<i>Advanced Micro Devices, Inc. v. LG Elecs., Inc.</i> , 14-cv-01012-SI, 2017 U.S. Dist. LEXIS 110776 (N.D. Cal. Jul. 17, 2017)	61
<i>Amdocs (Isr.), Ltd. v. Openet Telecom, Inc.</i> , 761 F.3d 1329 (Fed. Cir. 2014)	passim
<i>Drone Techs., Inc. v. Parrot S.A.</i> , 838 F.3d 1283 (Fed. Cir. 2016)	61
<i>Edwards Sys. Tech., Inc. v. Digital Control Sys., Inc.</i> , 99 F. App'x 911 (Fed. Cir. 2004)	42
<i>Griffin v. United Parcel Serv., Inc.</i> , 661 F.3d 216 (5th Cir. 2011)	24
<i>Hewlett Packard Co. v. Bausch & Lomb, Inc.</i> , 909 F.2d 1464 (Fed. Cir. 1990)	29
<i>Hilgraeve Corp. v. Symantec Corp.</i> , 265 F.3d 1336 (Fed. Cir. 2001)	34, 45
<i>In re Kollar</i> , 286 F.3d 1236 (Fed. Cir. 2002)	29
<i>In re Samsung Electronics Co.</i> , 22-mc-80005-VKD, 2022 U.S. Dist. LEXIS 25098, (N.D. Cal. Feb. 11, 2022)	25, 60
<i>Laserdynamics, Inc. v. Asus Computer, Int'l et al.</i> , 2:06-cv-348, 2009 U.S. Dist. LEXIS 3878 (E.D. Tex. Jan. 21, 2009)	61
<i>Lexion Med., LLC v. Northgate Techs., Inc.</i> , 641 F.3d 1352 (Fed. Cir. 2011)	23

<i>Linear Tech. Corp. v. ITC</i> , 566 F.3d 1049 (Fed. Cir. 2009)	35
<i>Liquid Dynamics Corp. v. Vaughn Co.</i> , 449 F.3d 1209 (Fed. Cir. 2006)	43, 59, 62
<i>MeadWestVaco Corp. v. Rexam Beauty & Closures, Inc.</i> , 731 F.3d 1258 (Fed. Cir. 2013)	42
<i>Media Rights Techs., Inc. v. Capital One Fin. Corp.</i> , 800 F.3d 1366 (Fed. Cir. 2015)	60
<i>Metro. Life Ins. Co. v. Bancorp Servs., L.L.C.</i> , 527 F.3d 1330 (Fed. Cir. 2008)	34, 35, 51, 53
<i>Nazomi Communs., Inc. v. Nokia Corp.</i> , 739 F.3d 1339 (Fed. Cir. 2014)	27
<i>Packet Intelligence LLC v. NetScout Sys.</i> , 965 F.3d 1299 (Fed. Cir. 2020)	28
<i>Versata Software, Inc. v. SAP Am., Inc.</i> , 717 F.3d 1255 (Fed. Cir. 2013)	passim
<i>Via Vadis Controlling GmbH v. Skype, Inc.</i> , 12-mc-193-RGA, 2013 U.S. Dist. LEXIS 23434 (D. Del. Feb. 21, 2013)	61
<i>Via Vadis, LLC v. Amazon.com, Inc.</i> , 14-cv-00813-LY, 2022 U.S. Dist. LEXIS 9169 (W.D. Tex. Jan. 18, 2022)	42

STATEMENT OF RELATED CASES

Two appeals from the same district court action on appeal here were previously before this Court: *In re Apple, Inc.*, 21-0187 (Fed. Cir.); *In re Apple Inc.*, 20-0104 (Fed. Cir.). No other cases pending in any other court may directly affect or be directly affected by this Court's decision in the pending appeal.

STATEMENT OF JURISDICTION

The district court had jurisdiction under 28 U.S.C. §§ 1331 and 1338. The district court entered a Final Judgment of Non-Infringement on June 29, 2023. Appx00001. Fintiv, Inc. timely filed a notice of appeal on July 27, 2023. Dkt.1. This Court has jurisdiction under 28 U.S.C. § 1295(a).

STATEMENT OF THE ISSUES

1. Whether the district court legally erred on summary judgment in requiring Fintiv to point to specific source code for the “widget” in Apple’s accused products when neither the claim, nor the claim construction of “widget,” requires source code?

2. Whether the district court legally erred on summary judgment by ignoring evidence presented by Fintiv that showed the existence of a widget and drawing inferences against Fintiv about the existence of a widget in Apple’s accused products?

STATEMENT OF THE CASE

The underlying patent infringement litigation was filed in December 2018 by Fintiv, Inc., asserting U.S. Patent No. 8,843,125 (the “’125 patent”) titled “System and Method for Managing Mobile Wallet and its Related Credentials.” Appx00259-00297. Fintiv specifically asserted that Apple, Inc. directly and indirectly infringed independent claims 11, 18, and 23 and dependent claims 12, 14, 20, 24, and 25 of the patent through Apple’s mobile payment and wallet applications found on Apple devices, such as the iPhone, iPad, Apple Watch, and Mac products. After 4.5 years of litigation, the district court overturned its prior denial of summary judgment and granted summary judgment of non-infringement to Apple, Inc. in a 10-page order. Appx00002-00011.¹ Fintiv now appeals the district court’s decision to overturn its earlier denial of summary judgment and issue a summary judgment of non-infringement in favor of Apple.

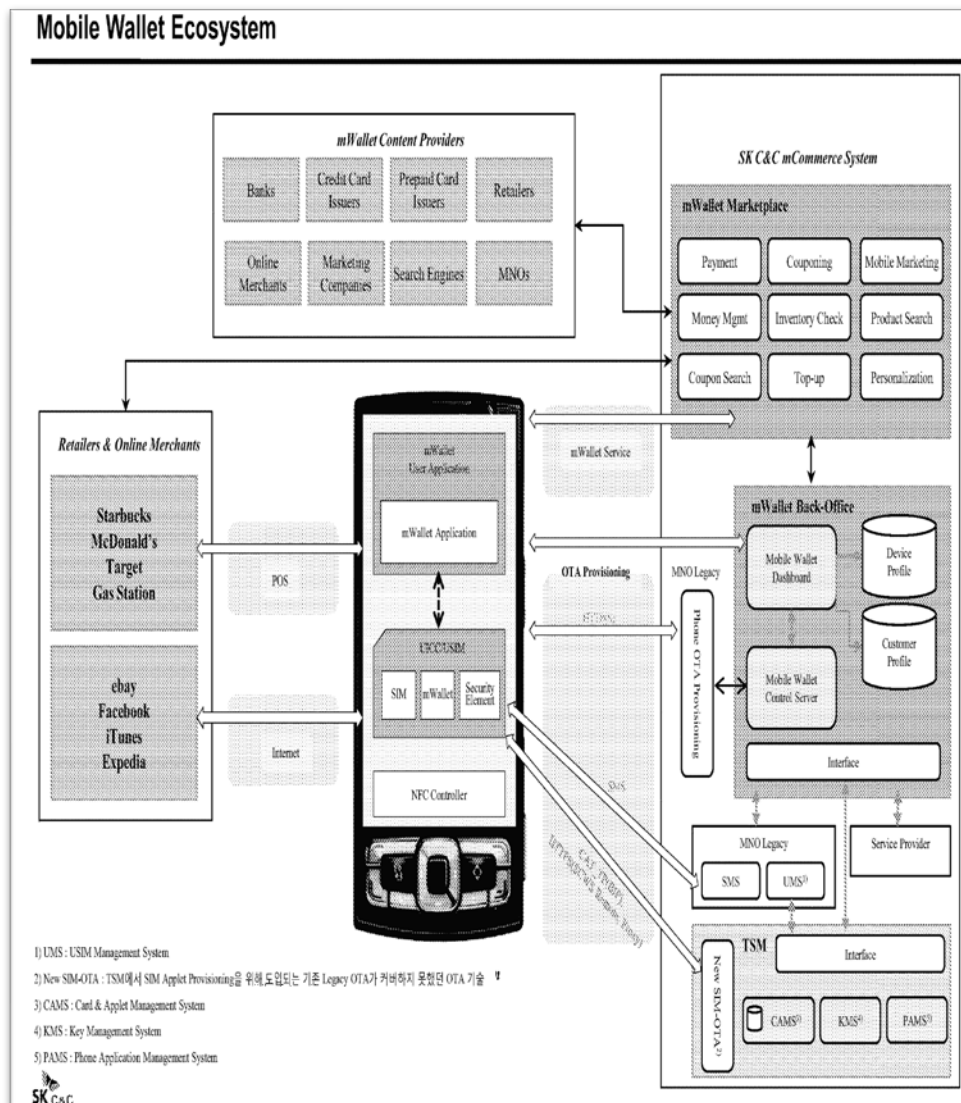
I. The Invention and the ’125 Patent

The invention at issue in this appeal was developed by the information and communications company, SK C&C (doing business as CorFire). Appx00085; Appx02052 (Business Requirements Document from ’846 provisional application); Appx26729-26730; Appx26901-26904 (Public Version); Appx26814; Appx26543-

¹ The district court’s sealed order is produced in the Appendix at Appx00002-00011. A public, redacted version of the order is produced in the Addendum and in the Appendix at Appx27316-27325.

26544. In and around 2010, SK C&C was developing a mobile payment ecosystem that could support devices and operating systems from multiple companies (*e.g.*, iPhone, Android, Windows Mobile, Blackberry, Palm-pre, etc.). In fact, from 2011 to 2012 CorFire employees met with Apple and disclosed the confidential and proprietary information behind SK C&C's mobile payment system, specifically SK C&C's wallet and widget technology. Appx 26729-26731; Appx26901-26904 (Public Version); Appx26729-26731.

The '125 patent itself originated from provisional applications filed by SK C&C in December 2010. Appx00085. As shown in the exemplary figure below, the system worked on mobile phones and implemented certain steps on network servers.



Appx01829 (Figure “Mobile Wallet Ecosystem”, ’846 Provisional Patent Application).

A. The Problems to be Solved

As noted in the ’125 patent, the idea of conducting credit card transactions between a mobile phone and a card reader was not new in 2010. Appx00092, 1:32-67. However, SK C&C’s invention, as claimed in the ’125 patent, solved two important problems. First, the industry at the time lacked any standardization of

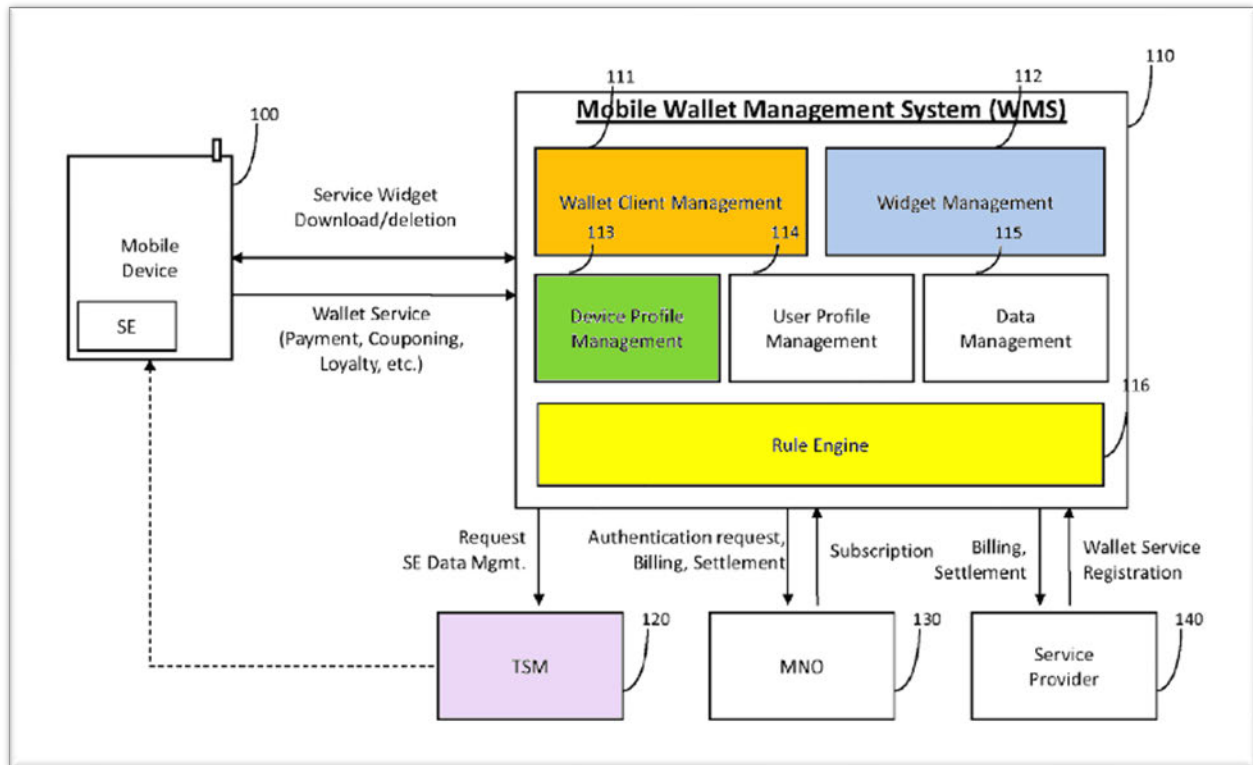
hardware or software; therefore, a user may try to install a Visa payment applet on their Blackberry device, only to later find that the Visa applet only worked with Samsung Android devices. Appx00092, 2:33-44; Appx00096, 10:48-49. Second, users had limited access to the account information stored on their mobile phones and “may be unable to view the details related to the contactless payment applets (*e.g.*, account number, expiration date, security code, balance and the like)” because industry security protocols required storage of the payment applet on special chips known as the “secure element” and would only allow “a limited generic description” of financial information to be displayed on mobile phones. Appx00092, 2:11-28.

B. The '125 Patent Solution

The '125 patent solves the problems of the prior art and describes an invention from both the server and mobile device perspectives. On the server side, the '125 patent describes and claims components of a “wallet management system” (WMS) generally shown in Figure 1 and recited in asserted claim 18. Appx00087; Appx00098, 14:7-23. On the mobile device side, the '125 patent describes and claims a “mobile wallet application” generally shown in Figure 2 and recited in asserted claims 11 and 23. Appx00088; Appx00098, 13:15-30, 14:38-53.

1. Server-Side Components

Figure 1 shows various components of the server side of the invention.



Appx00087 (highlighting added).

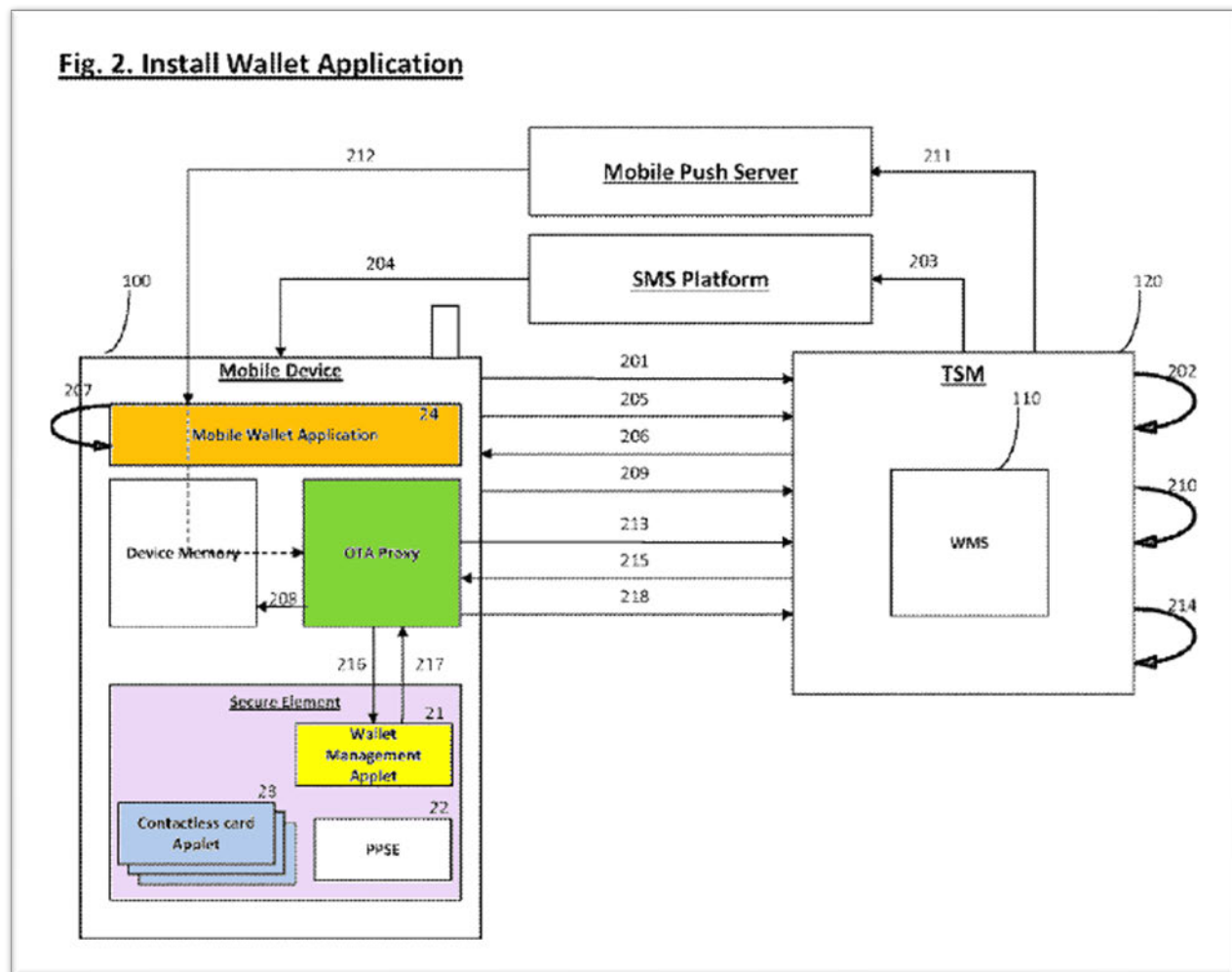
According to the patent, the server-side components solve the first problem by using the wallet management system (WMS) to filter the available applications and provide only compatible applications to mobile devices. Appx00096, 10:42-55.

Each component has a different function. For example, the Wallet Client Management Component 111 manages and stores information on the different types of wallet applications available for download, such as those “manufactured by Google®.” Appx00093, 4:61-67. The Device Profile Management Component 113

stores information related to the specific type of device and operating system. Appx00094, 5:9-16. The Rule Engine 116 provides the critical functionality of filtering the “mobile widget applications that are available for installation based on corresponding mobile device attributes” (*e.g.*, device type and operating system). Appx00096, 10:9-34. The Widget Management Component 112 “is responsible for the individual widgets” that are “configured to interface with a user of the mobile device.” Appx00094, 5:4-9. And, the Trust Services Manager (TSM) is a server that provides a consolidated point of contact for mobile devices to interact with third parties, such as network providers (*e.g.*, AT&T, Verizon), financial institutions (*e.g.*, Visa, Citibank), and mobile device manufacturers (*e.g.*, HTC, Motorola, Apple). Appx00094, 5:28-42; Appx00096, 10:18-34.

2. Mobile Device Components

Figure 2 of the patent shows the structure of the mobile device side of the invention.



Appx00088. The mobile device components solve the second problem associated with the prior art by providing a secure payment system within a secure element but also allowing the user to view their account information.

The mobile device system has multiple functions. The Mobile Wallet Application is similar to a traditional wallet with various payment or member cards that can be downloaded and stored on a mobile device. Appx00092, 1:43-46,

Appx00093, 4:61-67; Appx00094, 6:34-49. A Contactless Card Applet corresponding to each card in the wallet is stored in the secure element. The Secure Element stores sensitive information such as credit card account information that is used to make payments via contactless Near Field Communication (NFC). Appx00095, 8:23-28, 8:60-65; Appx00092, 1:51-62.

At the heart of the mobile platform is the Wallet Management Applet that solves the problem of viewing credit card information. The Wallet Management Applet stores a duplicate copy of the Contactless Card Applet account specific information. Appx00095-00096, 8:66-9:5. This duplicate is important because it is stored securely on the Secure Element, but unlike information in the Contactless Card Applet, the information in the Wallet Management Applet can be displayed to a user through a corresponding Widget in the mobile wallet application.

Widgets are applications that are typically configured with a user interface and reside in the Wallet Management Application at the application level to make them available to the user. Appx00094, 5:4-9, 5:66-6:4; Appx00095-00096, 8:63-9:5. Each widget corresponds to a virtual card and allows users to view and interact with it to obtain information about the card. *Id.*

3. The Representative Patent Claims

Fintiv asserted multiple claims of the '125 patent. The independent representative claims are 11, 18, and 23, which are set forth at the beginning of this brief.

II. The Accused Apple Products

A. Development of Apple's Wallet and Pay Products and Services

Apple's development of the accused Wallet and Apple Pay functionality in its products started years before they were ever launched. For instance, in 2011 and 2012, Apple looked to other companies to provide its mobile payment and wallet technology and met with the original owner of the '125 patent, SK C&C (d/b/a CorFire), to gather confidential and proprietary information about SK C&C's digital wallet and widget technology. Appx26729-26731 (attaching details and exemplary presentations at Appx26733-26783); Appx26901-26903. In fact, Apple later hired an employee of SK C&C as its Director of Apple Pay and Wallet Product Management. Appx26813-26814; Appx26730-26731; Appx26902-26903.

Not surprisingly, in 2014 Apple launched its Apple Pay and Wallet² products and services with the same wallet and widget functionality SK C&C disclosed to Apple in 2012. Appx26731; Appx26903; Appx18698. However, such functionality was claimed by SK C&C's '125 patent that Apple now infringes without license.

² Apple's wallet application was rebranded from Passbook to Wallet in 2015.

Specifically, Apple infringes the '125 patent by offering Wallet, Passbook, and/or Apple Pay functionality through its iPhone, Watch, iPad, Mac devices. Appx18686-18696. Apple further infringes through its use of network servers that enable Apple Pay functionality. Appx18696-18698.

B. Overview of Apple's Wallet and Apple Pay Products and Services

Apple Pay offers consumers a convenient way to make electronic payments with mobile applications, over the Internet, or by using near-field communication (NFC). Many credit card companies and merchants support and accept Apple Pay. Appx18698.

Consumers can use Apple Pay from applications, such as Apple's Wallet Application on their phone, Mac computer and/or Apple watch. First, the user "provisions" (or adds) the card to the Wallet Application. Appx18698-18699. To do so, the user adds their card information to the mobile device, manually or by image capture, which in turn is transmitted to Apple Servers. Apple's Wallet verifies the card information with the financial providers. Appx18758-18759; Appx19354.

The Apple Servers then transmit encrypted card information back to the device for storage in the Secure Element on that device. Appx18699; Appx18867; Appx20977-20978 ("The Apple Pay servers manage the setup and provisioning of credit, debit, transit, and Student ID cards in the Wallet app. The servers also manage the Device Account Numbers stored in the Secure Element."). During later

transactions, the device retrieves the encrypted card information and provides it to a merchant in encrypted form. Appx18699-18700. In this way, card details cannot be accessed by the transacting merchant and are kept private. Appx18700.

One convenience of Apple's Wallet and Apple Pay products and services, is that a user can maintain all payment cards in one place in Apple's Wallet. *Id.*



Appx18707.

[< Back](#)
[Next](#)

Card Details

Verify your card information.

Name

Card Number

Expiration Date

Security Code

[< Back](#)
[Next](#)

Upon selection via the “Next” and “Agree” buttons provided by the application, the device retrieves an applet to verify the card, *id.*, as shown below:

Terms and Conditions

Terms for Adding Your Chase Card to a Third Party Digital Wallet


These Terms for Adding Your Chase Card to a Third Party Digital Wallet (the "Terms") apply when you choose to add a Chase credit card, prepaid card or debit card ("Chase Card") to a digital wallet or other payment service managed or owned by a third party ("Wallet"). In these Terms, "you" and "your" refer to the cardholder of the Chase Card, and "we," "us," "our," and "Chase" refer to the issuer of your Chase Card, JPMorgan Chase Bank, N.A.

When you add a Chase Card to a Wallet, you agree to these Terms:

1. Adding Your Chase Card. You can add an eligible Chase Card to a Wallet by either following our instructions as they appear on a Chase proprietary platform (e.g., Chase Mobile® app or chase.com) or by following the instructions of the Wallet provider. Only Chase Cards that we determine are eligible can be

[Disagree](#)
[Agree](#)

Next



Card Verification

Choose how to verify your card for Apple Pay.

Email

Text Message

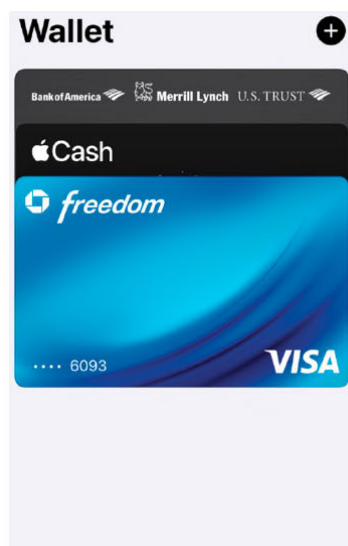
Text Message

Text Message

Call Chase

Complete Verification Later

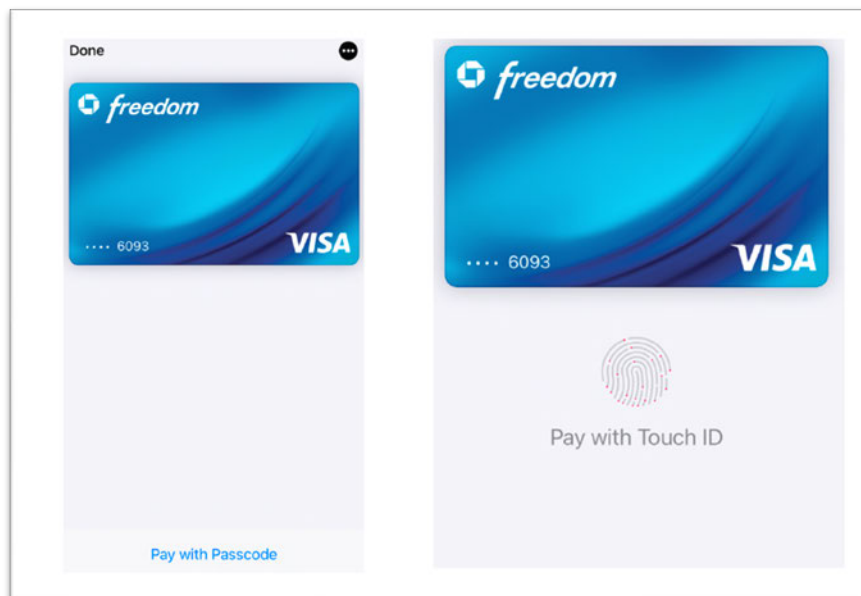
After receiving the contactless card applet, and a request for the card by the user, the Apple device retrieves a widget from the wallet management applet that corresponds to the user-selected debit or credit card. Appx18766-18768. When opened, Apple's Wallet shows the provisioned debit/credit card images by accessing the respective card widgets, which include the code for the card image and the underlying code that provides a user interface for each card. *See, e.g.,* Appx18767; Appx18790-18791; Appx18794-18795. The image below demonstrates Apple's Wallet with card widgets represented by card images for the user to select.



To use Apple's Wallet, a user selects a card within the wallet application. The widget not only provides the card image as a user interface but also provides routines a user can activate to perform transactions or view card information as shown in the various screenshots below. *See* Appx18767 ("...in connection with card

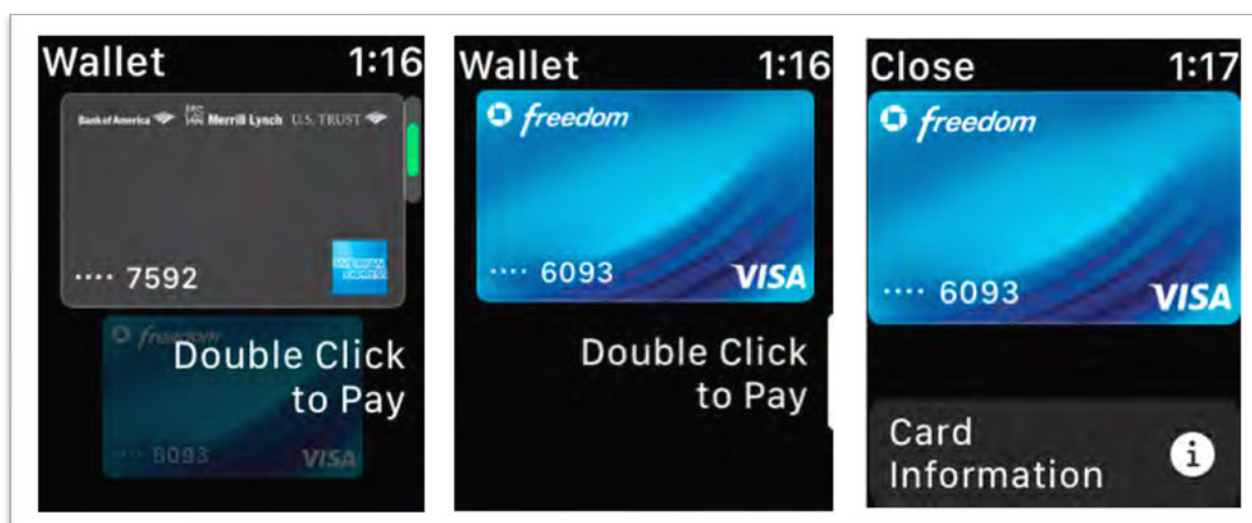
provisioning, a software (a “widget”) with a user interface and associated with a card being provisioned (e.g., a credit card) is retrieved.”); *see also* Appx18963-18964 (“[T]he interface knows that when you touch with the area ... that card art, code is invoked but then acquires...the payment instrument data for display...on the screen.”); Appx15247 (“[T]he widget is the code that’s sitting behind the—the card art that is activated when I touch it.”); Appx27375 (indicating the metadata code has to contain logic “because it has to include an indication of where to go to get the—the card credentials when the widget is invoked”).

As shown below, on the left, the user has selected the blue Visa from the wallet, and, in this example, they can choose to “Pay with Passcode” or “Pay with Touch ID.” Appx18497 (“To initiate a transaction with Apple Pay, a user provides a passcode or biometric confirmation (i.e. Touch ID or Face ID) to release the token from the secure element for processing.”).



Appx18791 (showing Apple’s Wallet functionality on an iPhone); *see also* Appx19322 (describing how “[u]sers can view their cards”); Appx19326 (same).

Widgets also provide code that can pull up additional details and card information when selected in the user interface as shown below.



Appx18794-18795 (showing Apple’s Wallet functionality on Apple Watch); *see also* Appx19201-19203 (showing “Latest Transaction” information).

III. The Proceedings Below

During that the litigation, Fintiv reviewed the source code and documents Apple produced, and Fintiv deposed multiple Apple witnesses. Fintiv’s software expert, Dr. Shamos, reviewed Apple’s documentation on the accused products, including the produced source code, and tested the accused products. On June 2, 2021, Fintiv produced Dr. Shamos’s 303-page expert report on infringement, Appx18648-18957, which concluded Apple infringed the asserted claims, including explaining that the evidence demonstrated that Apple’s products were used to retrieve, provision, store, manage, and filter a “widget” as the relevant claims required. Appx18658-18659; Appx18766-18788 (retrieving widgets); Appx18788-18806 (provisioning widgets); Appx18855-18857 (configured to store and manage widgets); Appx18859-18862 (configured to filter widgets); Appx18882-18894 (configured to store widgets); Appx18903-18913 (configured to provision a widget).

A. Claim Construction

On November 27, 2019, the Court issued its first claim construction order. Appx00051-00084. Among other terms, the court considered Apple’s request to construe the term “widget.” Appx00063-00067. Apple proposed that “widget” be construed to be a “user interface software application,” Appx00063, but the court rejected the construction because the construction sought to narrow the “widget” term to standalone applications, Appx00066. Instead, the patent specification made

clear that a “widget” could “reside in another application and ‘at the application level.’” *Id.* Thus, as the court noted, the “widget” could simply be “code, *e.g.*, a ‘plug-in,’ that runs within an application.” *Id.* The Court’s claim construction is:

[T]he Court construes “widget” as a plain-and-ordinary meaning and where the plain-and ordinary meaning is “software that is either an application or works with an application, and which may have a user interface.”

Appx00066.

B. Initial Summary Judgment Decision

On June 28, 2021, Apple filed a motion for summary judgment of non-infringement that contending there was no genuine issue of material fact that no “widget” existed in the accused products. In its briefing, Apple repeatedly questioned infringement on the basis that Fintiv’s expert admitted he could not point to an individual source code file that was the widget. Appx15171-15176. On September 23, 2021, the district court held a hearing on the summary judgment motion and other pre-trial matters, and denied Apple’s Summary Judgment motion without written opinion. Appx26184.

C. Reconsideration of Summary Judgment Decision

Almost two years after denying Apple’s motion for summary judgment of non-infringement, on June 7, 2023, the district court scheduled a pre-trial conference for Tuesday, June 13, 2023. Appx27231. On June 8, 2023, just three business days prior to the pre-trial conference, Apple requested to reargue the summary judgment

of non-infringement. The court scheduled that argument for the same day as the pre-trial conference – June 13, 2023. Appx27233. During the hearing, both parties presented their positions on summary judgment again, with Apple pushing the theory that summary judgment should be granted because Fintiv had not specifically identified the “widget” in Apple’s source code. Appx00005 (“Apple contends ... Fintiv identified no software code in the accused products that meets the ‘widget’ limitations”). By contrast, in its briefing, exhibits, and at the second hearing, Fintiv identified multiple sources of evidence indicating that the accused Wallet and Apple Pay products used “widgets.” The evidence included:

- Expert product testing and demonstration that identifies the widget from the card user interface shown in Apple’s Wallet application consisting of card art and underlying code that allows users to select virtual cards and perform certain functions, *e.g.*, view the card’s details, perform transactions, display latest transactions, or engage in other interactions with the widget. *See, e.g.*, Appx18767-18768 ¶¶309-310; Appx18790-18791 ¶¶359-360; Appx18882-18883 ¶578; Appx18928 ¶¶696-698; *see also* Appx18794-18795 ¶¶368-369; Appx19201-19203 (expert chart showing further device testing of widgets to display “Latest Transactions” information); Appx18961 (“[T]he interactivity that occurs between the user and that card information is performed by a widget.”); Appx18963-

18964 (“[T]he interface knows that when you touch with the area ... that card art, code is invoked but then acquires...the payment instrument data for display...on the screen.”); Appx15247 (“[T]he widget is the code that’s sitting behind ... the card art that is activated when I touch it.”); Appx27375 (“Q. Does the metadata in this case on the accused iPhone include logic? A. It has to because it has to include an indication of where to go to get the—the card credentials when the widget is invoked.”); *see also* Appx18656 ¶7 (identifying intention to bring demonstration of software based on report to trial); Appx25283 (exhibit list identifying devices to be introduced at trial demonstrating functionality).

- Source code files used for the purpose of performing actions upon widgets. *See, e.g.*, Appx18767 (retrieving widgets); Appx18855-18856 (configured to store and manage widgets); Appx18860 (configured to filter widgets); Appx18905 (configured to provision a widget); Appx18967-18968, 68:18-21; 69:20-22; Appx18981, 176:1-22.
- Technical Documents describing functionality that is a widget or uses a widget. Appx19317-19318 (showing retrieval of a user interface for card information); Appx19107; Appx19116-19120 (expert exhibits showing provisioning flow charts for widgets); Appx19322 (describing ability to view cards in wallet); Appx19326 (noting Wallet on Apple Watch allows

for viewing passes and cards details); Appx19330, Appx19105-19112; Appx19085-19090 (showing diagram to store and manage passes/widgets in expert chart); Appx19329 (describing digitizing cards and passes in Passbook with representation of credentials).

- Apple testimony about functionality matching the “widget” definition and use of that functionality. Appx19378-19380 (indicating passes could have a user interface and software that relates to the card); Appx19361, 130:11-23 (storage of passes).
- Source code files within Apple’s Wallet that identify a Code ID such as Code ID Appx18968; Appx19365-19366; Appx18612 (citing code file and directory structure for file at APPLE-FINTIV-SC_0242-243).³
- Other functionality similar to Apple’s Wallet and Apple Pay that Apple describes as a “widget.” Appx19307-19315 (showing weather widget and other widgets), Appx19333-19340 (same).

However, at the conclusion of the hearing, the district court reversed its initial summary judgment decision and granted summary judgment of non-infringement. Appx27298.

³ Due to confidentiality under protective order, source code files were filed with the briefing below, but are available upon request.

Fintiv pointed out that Apple’s argument did not seem like a summary judgment issue, but rather one of “sufficiency of the expert report, ... which is the first we’ve ever heard of that.” Appx27299. Fintiv, therefore, offered to submit a supplemental expert report, citing to the same documents, and to submit supplemental briefing. Rather than a dispute over the facts, Fintiv saw this as a “disagreement over the sufficiency of the expert report.” *Id.* The Court denied Fintiv’s request.

In its written order on June 21, 2023, the court held that Fintiv and its expert “failed to identify the claimed widget in the accused products” because Dr. Shamos conceded the source code files cited in his report were not themselves the claimed “widget.” Appx00006. Fintiv timely appealed.

SUMMARY OF THE ARGUMENT

Nothing requires Fintiv to specifically identify the location of “widget” source code to avoid summary judgment. This is particularly true where the claims, at most, only require code that acts upon a “widget” (*e.g.*, retrieves a widget or stores a widget) and not the “widget” itself. In this case, the court overemphasized the fact that Fintiv’s expert did not specifically identify “widget” source code, and the court ignored other evidence that widgets exist in the accused products (*e.g.*, screenshots of the widgets being used) as well as other circumstantial evidence of “widget” use in the accused products, including expert testimony, product manuals, and even

Wallet and Apple Pay code files that were described as using a Code ID [REDACTED] In view of the evidence that the accused products act upon a “widget,” the court erred in granting summary judgment.

Affirmance in this case would not only conflict with precedent on evidence necessary to survive summary judgment, it would unleash an unreasonable demand for source code in a multitude of cases. Source code review is already expensive and cumbersome to manage in any patent case, but to require or even prefer source code for every claim element lest the patent owner be subject to summary judgment, would force even more discovery and expensive review of source code when other evidence may be sufficient or even better.

Summary judgment of non-infringement should be reversed.

STANDARD OF REVIEW

Summary judgment is appropriate “if the movant shows there is no genuine issue as to any material fact and the movant is entitled to judgment as a matter of law.” Fed. R. Civ. P. 56(a). “This court reviews the district court’s grant or denial of summary judgment under the law of the regional circuit.” *Lexion Med., LLC v. Northgate Techs., Inc.*, 641 F.3d 1352, 1358 (Fed. Cir. 2011). Here, applying Fifth Circuit law, the Court reviews the district court’s decision to grant summary judgment *de novo*, applying the same standard as the district court.” *Absolute Software, Inc. v. Stealth Signal, Inc.*, 659 F.3d 1121, 1129 (Fed. Cir. 2011) (citation

omitted). The Fifth Circuit “views all evidence in the light most favorable to the non-moving party and draws all reasonable inferences in that party’s favor.” *Griffin v. United Parcel Serv., Inc.*, 661 F.3d 216, 221 (5th Cir. 2011).

ARGUMENT

The district court legally erred in granting summary judgment of non-infringement for at least two reasons. First, the court’s demand that Fintiv present direct evidence of “widget” source code conflicts, not only with the law, but also the asserted claim language. Second, even applying the district court’s understanding of the claims, if all inferences are drawn in favor of Fintiv, as required, there is at least a genuine issue of fact regarding the existence of a “widget” and Apple’s infringement.

I. The District Court Erred by Demanding Evidence of “Widget” Source Code

Apple’s demand that Fintiv present direct evidence of specific “widget” software code or source code is legally incorrect for two reasons. First, direct evidence of source code is not a requirement to prove infringement, even of computer-implemented claims. Second, the district court’s emphasis on “widget” code overlooked the actual claim language that—at most—only required evidence of code that performs certain actions upon the widget, *e.g.*, “*retrieving* a widget,” and not code for the widget itself.

A. Direct Evidence of Source Code is not Required to Prove Infringement

When considering accused products and patent claims implemented in computer code, this Court has been clear that patent owners “need not point to the specific location of the allegedly infringing code to overcome summary judgment.” *Amdocs (Isr.), Ltd. v. Openet Telecom, Inc.*, 761 F.3d 1329, 1341 (Fed. Cir. 2014); *see also Solas OLED Ltd. v. Samsung Elecs. Co. (In re Samsung Elecs. Co.)*, No. 22-mc-80005-VKD, 2022 U.S. Dist. LEXIS 25098, at *4 (N.D. Cal. Feb. 11, 2022) (“[A]ccess to source code may not always be required to show that a particular computer-implemented functionality exists....”). Rather, a patent owner is “entitled to establish genuine factual issues by relying upon its documentary evidence, without necessarily identifying the precise location of the allegedly infringing code.” *Amdocs*, 761 F.3d at 1343. In *Amdocs* for example, this Court rejected the defendant’s repeated argument that the patent owner “cannot prove infringement unless it analyse[d] DSD scripts and identifie[d] those that perform the claim limitations.” *Id.* at 1342-43. The Court chose instead to infer from marketing materials and other non-source code documentary evidence that there was a genuine issue of fact whether the defendant’s software infringed. *Id.*

In contrast to *Amdocs*, throughout its summary judgment order, the district court here overemphasized the importance of source code and rejected all other evidence as speculation if it was not the widget source code. Appx00006-00010.

The district court repeatedly criticized Fintiv’s failure to point to specific, direct evidence of code that represented the “widget” in the accused product:

- “But Fintiv’s opposition never identifies what ‘software’ or ‘software code’ comprises the claimed ‘widget’ in the accused products.” Appx00008; *accord* Appx00007 n.1; Appx00010.
- “Fintiv points to a source code module, [Code ID] [REDACTED] [Code ID] ... but never says that module is the claimed ‘widget’—because it is not.” Appx00008; *accord* Appx00006.

Apple and the court further reinforced their demand for source code evidence at the second summary judgment hearing:

- “If you want to go through it and take on why, for lack of a better word, the use of the credit card or applet doesn’t as a matter of law meet the Court’s definition here and it’s not software and *why it has to be source code*, if you would take that on now, and then, Mr. Waldrop, you can respond and keep going.” Appx27254 (emphasis added).
- “[W]here does he in his report explain what the code is that goes in and takes a look at the payment credentials so that the user can view them?” Appx27274.

Indeed, the district court put too much emphasis on the fact that Fintiv did not identify specific source code files that were the “widget.” *See, e.g.*, Appx00006-

CONFIDENTIAL MATERIAL OMITTED

00007; Appx00010; Appx27254; Appx27274.⁴ But, source code is simply a human-readable code used by computers to compile computer programs. It is not the actual machine code or application running on the computer. *See Nazomi Communs., Inc. v. Nokia Corp.*, 739 F.3d 1339, 1340 (Fed. Cir. 2014). Thus, the fact that a party does not point to a specific source code file for a computer application, does not mean that application running on a computer has no code.

The fact code exists can be proven, not only by the code itself, but also by the way a program operates or behaves when running. *See Versata Software, Inc. v. SAP Am., Inc.*, 717 F.3d 1255, 1261-62 (Fed. Cir. 2013) (holding that expert’s demonstration of the inherent functionality of the software was “the most telling evidence”). Such evidence is exactly what Fintiv’s expert used to opine on infringement. The fact that Dr. Shamos admitted he could not find a one single file that represented the “widget” source code, was not an admission that he did not identify evidence of a widget. To the contrary, Dr. Shamos directly identified the claimed widget in Apple’s accused devices through examples and screenshots of the product in operation. For example, he testified specifically that the “widget is represented by an icon showing card art, and when you touch that icon, software is

⁴ Fintiv’s expert did address Apple’s counsel’s attempts to force him to point to “one block of code that says this is the widget” because that is “not the way this is working;” they are in different components. Appx18963-18964; *infra* Section II.B.

invoked.” Appx18960; Appx18961; *see also e.g.*, Appx18767-18768 ¶¶309-310; Appx18790-18791 ¶¶359-360; Appx18882-18883 ¶578; Appx18928 ¶¶696-698; Appx19085-19087; Appx19201-19203 (expert chart showing device testing of widgets to display “Latest Transactions” information); *see also infra* Section II.

The court’s overemphasis on the need for direct evidence of source code, especially at the summary judgment stage, led the court to disregard all other expert or circumstantial evidence that the “widget” was an application or worked with an application on the accused Apple devices that had a user interface, as required by the “widget” construction. The district court’s demand for source code citations and failure to consider other evidence of a widget in the accused Apple devices was legal error and the judgment should be reversed and remanded for that reason alone.

B. The Demand for “Widget” Source Code Evidence Overlooks the Specific Claim Language

In addition to its unwarranted focus on a lack of specific source code citations, the district court erred by requiring evidence of “widget” code when the claims only require code that *acts upon* the “widget.” To the extent the district court is going to require direct evidence of specific source code, it must be sure it only requires code for what is specifically claimed. *See Amdocs*, 761 F.3d at 1343; *see also Packet Intelligence LLC v. NetScout Sys.*, 965 F.3d 1299, 1306 (Fed. Cir. 2020) (rejecting argument that jury did not have substantial evidence of “correlating connection flow entries” because claim language only required for “memory for *storing* flow entries”

and not correlating (emphasis-in-original)). In *Amdocs*, the Court reversed summary judgment where the district court indicated there was no evidence the accused system could “*generate* output records ‘close to the source’ of the network information.” *Id.* (emphasis added). As the Court noted, the *Amdocs* district court erred because nothing in the claims required the system to “*generate* ... records close to the source.” *Id.* (emphasis added). Instead, the claims required “only *enhancement* to occur ‘close to the source’ of the network records.” *Id.* at 1343 (emphasis in original).

Here, as in *Amdocs*, the district court granted summary judgment holding Fintiv had pointed to “no software in the accused products that constitutes a ‘widget.’” Appx00004. But like *Amdocs*, this conclusion is based on a misunderstanding of what the claim language specifically requires. The asserted claims do not require widget code, rather, they are method or configuration claims that require performing *actions* on a widget, and for those claimed actions, Fintiv disclosed ample code.⁵ See *infra* Section II. For example, Claim 11 recites a method that includes steps for “**retrieving** a widget” and “**provisioning** ...the widget.”

⁵ Apparatus or product claims cover what a product *is*, method or process claims cover what a product *does*. See e.g., *Hewlett Packard Co. v. Bausch & Lomb, Inc.*, 909 F.2d 1464, 1468 (Fed. Cir. 1990) (referring to apparatus claims); *In re Kollar*, 286 F.3d 1327, 1332 (Fed. Cir. 2002) (noting that “[a] process [or method] ... consists of acts, rather than a tangible item).

Claim 18 requires code for “a widget management component configured to **store** and **manage** widgets” and a rule engine to “**filter** a widget.” Claim 23 requires code for “a mobile wallet application configured to **store** a widget” and a “proxy configured to **provision** ... a widget.” No asserted claim requires the accused device to comprise simply “a widget.” Thus, even if Fintiv must provide direct evidence of the source code implementing the claim limitations (which is not the law), it would only need to provide evidence of the code explicitly called for in the claims, *e.g.*, code for “retrieving” the widget, not the widget by itself. *See Amdocs*, 761 F.3d at 1343.

The district court’s complaint that the source code files cited in Dr. Shamos’s report were admittedly not the “widget” code was error. Dr. Shamos was intentionally pointing to the code for steps of acting on the widget, required by the claims. *See, e.g.*, Appx18767 (retrieving widgets); Appx18855-18856 (configured to store and manage widgets); Appx18860 (configured to filter widgets); Appx18905 (configured to provision a widget); Appx18967-18968, 68:18-21, 69:20-22; Appx18981, 176:1-22.

The error in the district court’s decision is further highlighted by its repeated requirement that the “widget” must be “software *in the accused products*.” Appx00004 (emphasis added); *see also* Appx00006 (titling its heading “Fintiv And Its Expert Failed To Identify The Claimed ‘Widget’ In The Accused Products”);

Appx00007 (criticizing as “speculation that there must be a ‘widget’ in the products”); Appx00010 (“nowhere does Fintiv’s opposition state that ‘the “widget” in the accused product is X,””). But this makes no sense. The accused software infringes the method and configuration claims because it performs an *action* on a “widget” (e.g., retrieving, provisioning, etc.), and not because the software *is itself* the “widget.” Appx00098. Logically, the infringing code for “retrieving a widget” cannot also be the “widget.” Thus, it is error for the district court to require direct evidence of actual widget code to satisfy the limitation that, at most, only requires code for acting upon widgets.

To be sure, the distinction between code for acting on a widget and the code for the widget itself is not a trivial one. In its order, the district court discounted Fintiv’s expert, Dr. Shamos, because he “had Apple’s source code, but still found no ‘widget.’” Appx00007 n.1.⁶ However, this is to be expected as the widget does not necessarily reside in Apple’s source code for the wallet management application. As Apple readily admitted during its transfer hearing, third parties supply the widget for the product—not Apple. Appx01595-01596 (“NXP gives us a turn key product that has the terms that has the *widget*, the apps, the applet in it.” (emphasis added)).

⁶ Per the restrictions on the Protective Order, Dr. Shamos never “had” Apple’s source code; he was able to examine it and print a limited number of pages.

Though it is certainly true that there must be some direct or circumstantial evidence that the accused product code acts upon widgets, it is error to demand that Fintiv must show a widget in Apple’s own source code. *See Amdocs* 761 F.3d at 1343 (reversing summary judgment where factual dispute over location of code).

For example, if a hypothetical patent claimed software for “retrieving a photo,” and an accused infringer sold photo management software with code for retrieving a photo, the patentee should not be required to point to both the infringer’s retrieval code and the code for photo itself. In that case, it would be a third party (*e.g.*, the infringer’s customers) that supply the photos. While the patent owner may need to show from marketing documents or other product documents that the purpose of the accused software was to operate on photos, actually demanding evidence of the customer’s photo code is an evidentiary step that goes too far.

Thus, the district court erred by requiring evidence of “widget” code within Apple’s accused product, when at most, the claims only require code that acts upon widgets within Apple’s products.

C. The Claim Construction of “Widget” Does Not Require “Widget” Code

Undoubtedly, part of the district court’s intense focus on widget code appears to come from the district court’s construction of the “widget” term. Though the court gave “widget” its plain-and-ordinary meaning, it clarified that the plain-and-ordinary meaning required a “widget” to be “software that is either an application or

works with an application, and which may have a user interface.” Appx00067; Appx00084. But, defining “widget” as software does not convert the evidentiary burden to one that requires putting into evidence a source code file that is the “widget.” *See, e.g., Amdocs*, 761 F.3d at 1343 (holding that “the fact that the parties dispute the code’s location does not mean ... that Amdocs cannot prove infringement as a matter of law”); *Versata*, 717 F.3d at 1262 (holding that, whether source code met claim language of “computer instruction,” was a fact issue properly resolved by expert testimony, and concluding it was). Based on the construction, the district court seems to have conflated code for the widget with code for the accused product application. A “widget” may be software that is an application or works with an application, but it does not change the fact that the claims only require actions upon that software and not the widget software itself.

Furthermore, there is nothing in the construction that would change Fintiv’s evidentiary burden and require direct evidence of actual source code. Though the construction requires a “widget” to be software that may have a user interface, circumstantial or expert evidence of the existence of that widget through other interaction with it, and discussion of the accused devices’ interaction with it is enough to survive summary judgment. *See Versata*, 717 F.3d at 1261 (approving expert demonstration of functionality as “most telling”). In fact, the court’s construction never defines the widget as code or source code in the first place.

Apple’s argument that there must be a source code file that is the “widget” simply attempts to rehash the arguments that the court rejected during claim construction. As the district court recognized at claim construction, the widget does not have to be a stand-alone application in any sense, but can simply work with an application or within it. Appx00066. Thus, there is no requirement a widget be a separate file of some sort.

II. Even with the District Court’s Requirement of Code, Genuine Issues of Fact Confirm the Existence of a “Widget”

Setting aside the district court’s misunderstanding of what the claims actually require, the court still erred by finding there was no evidence of a “widget” in the accused products. “[S]ummary judgment of non-infringement can only be granted if, after viewing the alleged facts in the light most favorable to the non-movant, there is no genuine issue whether the accused device is encompassed by the claims.” *Hilgraeve Corp. v. Symantec Corp.*, 265 F.3d 1336, 1341 (Fed. Cir. 2001) (citation omitted). Where the facts and interpretations of evidence are disputed, a court cannot make credibility determinations or weigh the evidence, but instead must draw all legitimate inferences in favor of the non-moving party. *See Metro. Life Ins. Co. v. Bancorp Servs., L.L.C.*, 527 F.3d 1330, 1338-39 (Fed. Cir. 2008). In determining infringement, an accused product may infringe “if it is reasonably capable of satisfying the claim limitations.” *Hilgraeve Corp.*, 265 F.3d at 1343.

Thus, regardless of whether Fintiv must provide evidence of “widget” code in the accused products as Apple proposes, to survive summary judgment, Fintiv was only required to present evidence that created a legitimate inference of the existence of a widget that was reasonably capable of satisfying the claim limitations. Such evidence does not have to be direct evidence. To prove infringement, “a patentee must show that a defendant has practiced each and every element of the claimed invention, and may do so by relying on either direct or circumstantial evidence.” *Linear Tech. Corp. v. ITC*, 566 F.3d 1049, 1060-61 (Fed. Cir. 2009) (citation and internal quotation marks omitted).

Among other things, this Court has approved the use of product documentation, expert testimony, demonstrative examples, and prior art comparisons as sufficient sources of evidence to determine if claim limitations are met. *Amdocs*, 761 F.3d at 1343 (“Amdocs is entitled to establish genuine factual issues by relying upon its documentary evidence, without necessarily identifying the precise location of the allegedly infringing code.”); *Versata*, 717 F.3d at 1261, 1262-63 (noting expert demonstration evidence of software functionality that inherently met the claim limitation was “[t]he most telling evidence” and approving comparisons of accused products to prior art to prove infringement); *Metro. Life Ins.*, 527 F.3d at 1338-39 (reversing grant of summary judgment where expert provided opinion based on the combined examination of “source code, various Metlife

documents, and particular spreadsheets”). Here, Fintiv has presented the same types of evidence to prove the existence of a widget, and it was error for the district court to discount that evidence and grant summary judgment.

A. Fintiv and its Expert have Specifically Identified the “Widget” Software Through Tests and Examples from the Accused Products

1. Dr. Shamos Demonstrated how the Functionality of Apple’s Wallet and Apple Pay Used the Claimed “Widget”

This Court has been clear that an expert may prove infringement by demonstrating the functionality of a piece of software through its operation. *Versata*, 717 F.3d at 1262-63. In *Versata*, for example, the court affirmed a jury verdict of infringement noting that “[t]he most telling evidence was the expert’s demonstrative data setup” in which the “expert used the inherent functionality of [the accused] software” to prove how certain claim limitations were met. *Id.* at 1261.

Moreover, like the “widget” term here, the SAP defendant in *Versata* argued that Versata provided no evidence that the “accused software used denormalized numbers during run-time.” *Id.* at 1263. However, the Court held there was sufficient evidence to support a jury verdict because the expert testified the accused software contained numbers without “fixed units” and “the numbers can assume a different meaning depending on which pricing operation is being performed” as the agreed definition of “denormalized number” required. In his testimony, the Versata expert

compared prior art that used “fixed units” to the accused software that did not and inferred the accused software must inherently look to other information to associate units and information during run time. *Id.* at 1263.

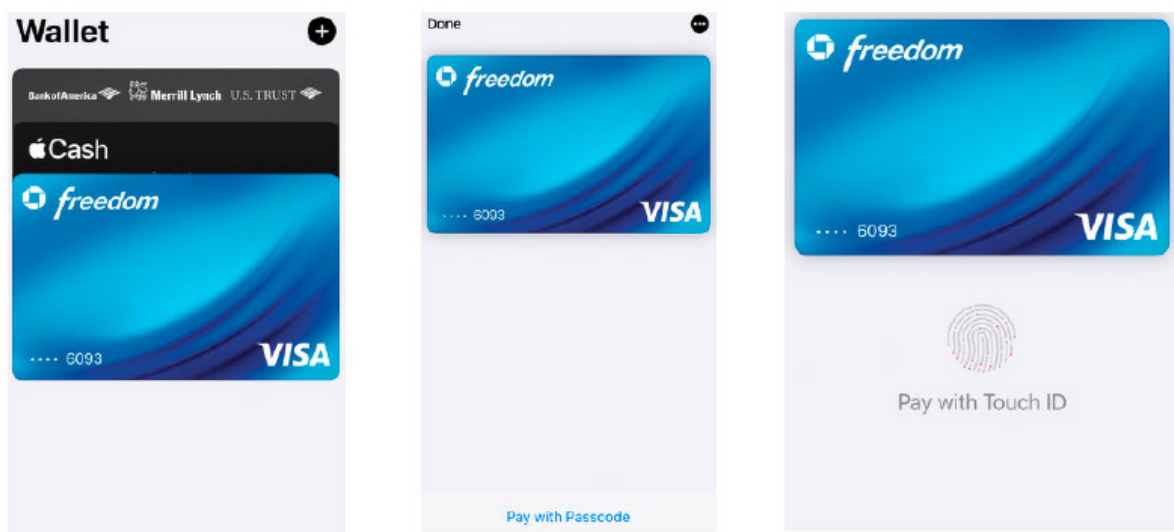
As in *Versata*, Fintiv’s expert, Dr. Shamos, repeatedly described how the functionality exhibited in Apple’s Wallet and Apple Pay products used “widgets” that met the claim limitation.

First, Fintiv’s expert provided a description of where the widget functionality was located, from where the widgets were retrieved, and what functions the widgets performed in the accused product. Dr. Shamos opined that the widget in the accused product “allows a user to, for example, *view the card’s details or perform transactions,*” that “(the widget associated with the credit card) also works with another application (*e.g., the Wallet Application that presents multiple widgets that a user can use,*” and that those “widgets ..., including their associated software code such as card image code and metadata code, are retrieved from servers (*e.g., Apple’s Code ID* servers) and device memory.” Appx18767 ¶309 (emphasis added); *see also, e.g.,* Appx15247, 78:4-14; Appx18961; Appx18963-18964; Appx18984-18985 (discussing ¶309); Appx18790-18791 ¶¶359-360; Appx18882-18884 ¶¶578-580; Appx18927-18928 ¶¶696-698; Appx19085-19087; Appx19201-19203 (expert claim chart showing further device testing of widgets to display “Latest Transactions” information).

CONFIDENTIAL MATERIAL OMITTED

Second, Dr. Shamos went further to describe testing of the product and provide multiple exemplary screenshots from the Apple products that show the user interfaces of the accused widgets in the form of credit cards. Dr. Shamos explained that the screenshots provided specific widget examples for various credit cards in Apple's Wallet that included the card image, a user interface that allowed the user to select the card, and exemplary actions that could be performed by the widget software once selected (e.g., viewing the card details and performing transactions with the card).

359. For instance, the widget (providing a user interface) is also provisioned (made available for use), as reflected in the screenshots below. Each screenshot below (showing the virtual card image for the Visa card) presents a software (with a user interface) that is made available to the user for selecting, via its user interface, among the available ones to, for example, view the card's details or perform transactions.

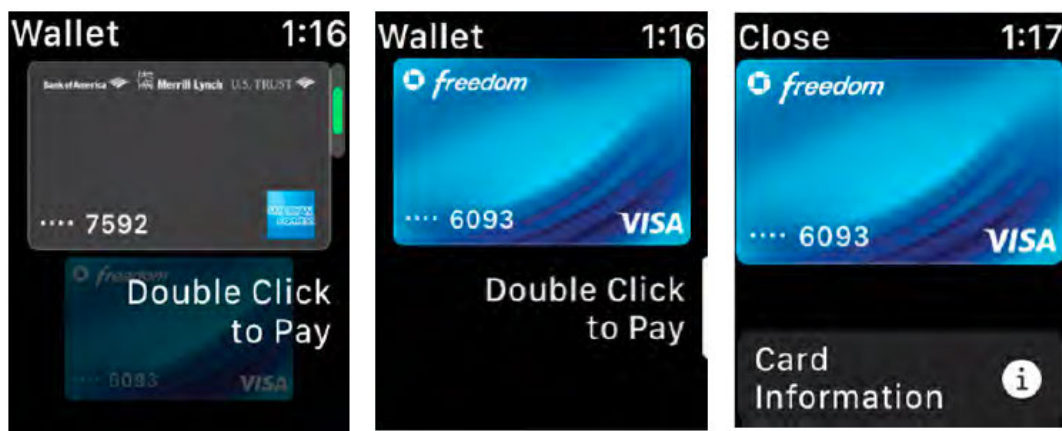


360. Alternatively, the software (the widget associated with the Visa card) also works with another application (e.g., the Wallet

Application) that presents multiple widgets that a user can use, as shown above.

Appx18790-18791 ¶¶359-360.

368. Further, the widget (providing a user interface) is also provisioned (made available for use), as reflected in the screenshots below. Each screenshot below (showing the virtual card image for the Visa card) presents a software (with a user interface) that is made available to the user for selecting, via its user interface, among the available ones to, for example, view the card's details or perform transactions.



369. Alternatively, the software (the widget associated with the Visa card) also works with another application (e.g., the Wallet Application) that presents multiple widgets that a user can use, as shown above.

Appx18794-18795 ¶¶368-369; *see also* Appx18882-18884 ¶¶578-579; Appx18885-18886 ¶¶584-586 (providing screenshots and additional examples of the widgets); Appx19086-19087, Appx19201-19203 (expert chart showing additional widget examples).

Third, under examination at his deposition, Dr. Shamos stood by his report and reiterated where the widget was in the accused product as shown by the functionality exhibited in the screenshot examples.

For example, I can select it to -- as a payment instrument or I can select it and view its details or I can select it and I can delete it. And so the interactivity that occurs between the user and that card information is performed by a widget.

* * *

So the -- the widget is software that enables you to do things with the card.

Appx18961.

So the widget consists of a bunch of things. One is that there is a -- a screen manager and it manages objects on the screen. One of those objects that's on the screen is a -- is card art that corresponds to a particular payment instrument.

And the interface knows that when you touch within the area of that -- that card art, code is invoked but then acquires the device -- the payment instrument data for display on the -- for display on the screen.

So that code is a combination of pre-existing Apple operating system code that has been specially configured during provisioning so that it understands this particular -- where to get the data for this particular payment instrument.

Appx18963-18964.

So the widget is software. Software can have multiple components, some of which call each other, et cetera. And so there is no concept of this thing called a widget that -- that everything is in. They're different -- they're different components.

The -- the code that recognizes the geographic coordinates of a touch on the screen is pre-existing in the Apple device because then

that's how you interact with the Apple device is by -- is by touching the screen.

Now, what happens when you touch a particular place, that changes depending on which widgets have been installed. That is, how the interface has been configured to deal with a particular card.

Before I provision a card there is nothing on the screen I can touch to get access to that card. Once I have provisioned a card, then the icon exists and the underlying code exists that retrieves the credentials associated with that card.

Appx18965-18966.

[T]he widget is the code that's sitting behind the -- the card art that is activated when I touch it.

Appx15247.

Q. Does the metadata in this case on the accused iPhone include logic?

A. It has to because it has to include an indication of where to go to get the -- the card credentials when the widget is invoked.

Appx27375.

Q. So doesn't -- wouldn't the widget have to have something done to it in order to make it available for use?

A. Yeah, it has to be installed on the machine. It wasn't there before. After I provisioned, it's there. And it's available for use after I verified my card with my issuer, because I can make payments with it and I can delete it and do other things to it.

Appx18970.

Q. Where -- following provisioning, where is the widget stored on the iPhone?

A. I don't know which memory it sits in, but clearly it's there because when I turn on my phone it pops up. And that's true whether I have an internet connection or not.

Appx18970.

While Apple may disagree with Dr. Shamos's analysis and present its own opposing witness's opinions on whether the products or the functionality exhibited can be a widget, such a battle of the experts is properly presented to a jury and not resolved on summary judgment. *See MeadWestVaco Corp. v. Rexam Beauty & Closures, Inc.*, 731 F.3d 1258, 1269 (Fed. Cir. 2013) (refusing to overturn infringement finding that turned on credibility determination in battle of experts); *Via Vadis, LLC v. Amazon.com, Inc.*, 14-cv-00813-LY, 2022 U.S. Dist. LEXIS 9169, *22 (W.D. Tex. Jan. 18, 2022) (citing *Edwards Sys. Tech., Inc. v. Digital Control Sys., Inc.*, 99 F. App'x 911, 921 (Fed. Cir. 2004) (nonprecedential)). At this stage, based on expert analysis of the product and its operation, Fintiv is entitled to the same inference provided in *Versata*, 717 F.3d at 1261-63. That is, Fintiv is entitled to an inference that the functionality demonstrated by Dr. Shamos through the card images and underlying code for each card in the wallet application obviously creates an interface for users to obtain card information and perform transactions that would be "widgets" as construed by the court, *i.e.*, "software that is either an application or works with an application, and which may have a user interface." Appx00067.

2. Circumstantial Evidence Indicates that Apple's Accused Products Use "Widgets"

In addition to simply observing the product functionality and offering expert opinion on that functionality, Fintiv presented additional circumstantial evidence that widgets are used in the accused products. Circumstantial evidence is more than sufficient, and in some cases even more certain than direct evidence, as a means to prove infringement. *See Liquid Dynamics Corp. v. Vaughn Co.*, 449 F.3d 1209, 1219-20 (Fed. Cir. 2006). In *Liquid Dynamics*, for example, this Court examined evidence of slurry tanks accused of infringing patent claims that required specific placement of flow nozzles. *Id.* Although there was no direct evidence that certain structural limitations were met for 11 tanks, and there was even direct evidence that 4 of the 11 tanks did not meet the structural limitations, this Court held that it was still reasonable for a jury to rely upon general information in an engineering manual and records from other similar tank installations to infer that the 11 tanks met the limitation. *Id.*

In this matter, Dr. Shamos identified numerous source code files responsible for acting upon widgets. Appx18767 ¶309. For example, Dr. Shamos identified the [Code ID] file as used to retrieve widgets and described it as "essential to getting the widget on the machine." Appx18967. He further identified the [Code ID] file and others as "involved in the creation of the widget." Appx18968; *see also* Appx18982-18984. He identified [Code ID] files as used to provision the widget. Appx18905

¶630. He identified files used to filter widgets. Appx18860 ¶528. And further, he identified the files that store and manage widgets. Appx18855-18856 ¶¶518-519.

To be sure, Fintiv even presented evidence in the source code file structure for Apple’s Wallet showing that Apple itself considered the payment source code to act upon “widgets.” In one such file Code ID the name actually referred to its functionality as providing a Code ID for the Wallet Application. Appx18612 (citing code file at APPLE-FINTIV-SC_0242-243); Appx19365-19366; Appx18968, 69:10-14. Notes and comments in source code are viable forms of infringement evidence. *Versata*, 717 F.3d at 1261 (approving expert reliance on “notes or comments” in the source code for evidence of infringement).

Although Dr. Shamos agreed that the source code modules he identified were not the “widget” themselves, he testified that the modules “indicate the presence of widgets in the accused Apple devices.” Appx18983. As explained previously, Fintiv presented the expert analysis in ¶309 and elsewhere to identify code for “retrieving” the widget or performing other actions on widgets—not the code for the widget itself. *See* Appx18766-18768; *see also* Appx18606-18607. The fact that Dr. Shamos did not point to a source code file for a widget actually retrieved or created by the code is not dispositive because this Court does not require actual operation of the code to establish infringement, *see Amdocs*, 761 F.3d at 1342-43,

CONFIDENTIAL MATERIAL OMITTED

and only requires evidence that an accused product is “reasonably capable of satisfying the claim limitations.” *Hillgraeve*, 265 F.3d at 1343 (internal citations omitted).

Reading this testimony in a light most favorable to Fintiv, if there are files for retrieving, building, provisioning, storing, managing, and filtering widgets, as Dr. Shamos identified, and Apple specifically refers to these files as acting on a [Code ID] to provide a [Code ID]; then it is more than reasonable to infer that those files do indeed retrieve, build, store, manage, filter, and [Code ID] “widgets.” Indeed, the source code functionality further confirms what Dr. Shamos’s identified in the screenshots—[Code ID] of widgets that have been retrieved, provisioned, stored, and managed. *See infra* Section II.B.

Further technical documents identified by Fintiv provide even more evidence that Apple retrieves and provisions widgets in Apple Pay. For example, Apple’s technical documents titled show requirements for provisioning including code that retrieves a user interface for card information.

CONFIDENTIAL MATERIAL OMITTED

Apple Tech Image



Appx19317-19318. This user interface is the same card information identified in Dr. Shamos's screenshots as the "widget" and one of the explicit software functionalities the construction of "widget" adopts. Appx00067. Thus it is reasonable to infer that retrieval of user interfaces that are provisioned to display card information, are the same user interfaces that will constitute a "widget" as widgets were explicitly defined by the district court to include software with a user interface.

B. The District Court's Criticisms of Fintiv's Evidence Failed to Consider the Proper Context, Much Less Overcome an Inference in Favor of Fintiv

Throughout its order, the district court considered the evidence presented, but failed to view the evidence in the proper context or with the proper inference.

CONFIDENTIAL MATERIAL OMITTED

1. Dr. Shamos’s Statement that he did not Identify a Specific “Widget” Source Code File was Not a Concession that there was No Widget

First, the district court criticized Fintiv’s expert for conceding that source code files cited in his report were not the source code for the widget. Appx00006-00010. But, patent owners are not required to present direct evidence of source code for every limitation of a computer-implemented claim. *See Amdocs*, 761 F.3d at 1341-1343; *Versata*, 717 F.3d at 1261-63 (relying on demonstrations of code functionality to establish infringement of certain elements). In his deposition testimony, Dr. Shamos admitted that the source code files he listed were not themselves the “widget,” but that is not the same as an admission that he had no evidence of widget software. As previously noted, the claims require code that act *upon* widgets to perform retrieval, provisioning, and other actions; therefore, Dr. Shamos pointed to that code specifically in his report. *See, e.g.*, Appx18767-18768 ¶309; *see also supra* Arg. Section I.A., II.A.

While Apple and the district court are critical of Dr. Shamos’s testimony that those source code files were not by themselves an entire “widget,” there was never a representation that they were. Appx18963-18967. In reality, Dr. Shamos identified the widget software by pointing to screenshots of card interactivity functionality provided to a user in the accused product that demonstrated underlying

software and a user interface that would constitute a “widget.” Appx18790-18791 ¶¶359-360; Appx18961, 45:7-19.

In fact, reviewing in context Dr. Shamos’s testimony, reveals how Apple and the district court misconstrued the expert testimony and failed to accord it the proper inferences. For example, Dr. Shamos explains that files, such as Code ID cannot be all of the widget and attempts by Apple to find “one block of code that says this is the widget” or “a routine that would be, for example, widget.m [is] not the way this is working.” Appx18963-18964. As Dr. Shamos explained,

So the widget is software. Software can have multiple components, some of which call each other, et cetera. And so there is no concept of this thing called a widget that -- that everything is in. They're different -- they're different components.

The -- the code that recognizes the geographic coordinates of a touch on the screen is pre-existing in the Apple device because then that’s how you interact with the Apple device is by -- is by touching the screen.

Now, what happens when you touch a particular place, that changes depending on which widgets have been installed. That is, how the interface has been configured to deal with a particular card.

Before I provision a card there is nothing on the screen I can touch to get access to that card. Once I have provisioned a card, then the icon exists and the underlying code exists that retrieves the credentials associated with that card.

Appx18965-18966.

Rather than give the proper credit and inference to this testimony, the district court simply ignored it and incorrectly found that the widget software code must be

CONFIDENTIAL MATERIAL OMITTED

represented in a single source code file. For example, the district court found that Dr. Shamos “confirmed that none of those files is a ‘widget’” including the [Code ID] file. Appx00006. But, the testimony did not confirm absence of a widget. Given the proper inference, it only confirmed that a widget was present, as Dr. Shamos explained that [Code ID] *alone* can’t be the widget It can’t be *all* of the widget,” Appx18963 (emphasis added), but it would still be part of the widget across multiple files. Appx18965-18966. At most, the testimony simply suggests that the widget may not be in a single file, but it did not confirm there was no widget or that the [Code ID] file was not evidence of a widget. To the contrary, Dr. Shamos’s testimony confirms that the [Code ID] file, along with the [Code ID] and [Code ID] files, Appx18968, are evidence of a widget.

The district court’s demand for a single source code “widget” file is not supported by the claim construction or the patent specification. In fact, the patent specifically notes that the “widget may reside in the mobile wallet application 24, at the application level, to provide an interface to the user,” Appx00095, 8:63-65, but it says nothing about the widget being created from a single source code file.

CONFIDENTIAL MATERIAL OMITTED

2. The District Court Erred in Concluding that no Expert Personally Reviewed the Apple Source Code to Support Dr. Shamos's Opinions

Next, the district court criticized the fact that Dr. Shamos cited “no deposition testimony nor any expert who ‘personally reviewed’ the Apple source code to support his speculation that there must be a ‘widget’ in the products.” Appx00007. But this makes no sense for multiple reasons.

First, Dr. Shamos *is the expert* that reviewed Apple's source code and determined that the Accused Products used widgets. Appx18659. Indeed, not only did he provide examples of his testing in the screenshots, but he identified the code files for retrieving, creating, provisioning, storing, managing, filtering, and viewing the widgets in the product. From that, it was reasonable for him to conclude—as he did—that the accused products use widgets that meet the claim limitations. Appx18985-18986.

Second, Dr. Shamos was not simply speculating as to the existence of a widget. He presented examples of widgets (*e.g.*, credit cards) on Apple devices that used software to provide users with a user interface that could provide card information and perform transactions. *See, e.g.*, Appx18767, Appx18790-18791; Appx18882-18883; Appx18928; Appx18794-18795; Appx19085-19087; Appx19201-19203. These are the same examples of “widgets” identified in the patent specification as “an application configured to interface with a user of the

mobile device. In an example, widgets may refer to individual payment applications, transportation applications, and other related applications.” Appx00094, 5:6-9.

Dr. Shamos further identified specific code that acted on or performed actions related to those widgets and even used relevant naming conventions. Appx19365-19366; Appx18968; Appx18612 (citing APPLE-FINTIV-SC_0242-243). Though he did not identify a single, specific source code file for the widget, his opinions were certainly not based on speculation as the district court found. Appx00007-00010. He is an expert in computer technology and electronic payment systems, Appx18657-18658, and his tests of the accused applications and review of Apple documentation and code to conclude that there is widget software implemented in the accused product was proper based on that evidence. *See Versata Software*, 717 F.3d at 1261-63; *Amdocs*, 761 F.3d at 1341-43.

If Apple does not agree with those interpretations and analysis, the proper way to challenge them is through cross examination or interpretations of its own expert and not summary judgment. The district court’s decision to discredit his opinion and the circumstantial evidence that widgets exist was error. *See Metro. Life Ins.*, 527 F.3d at 1338-39 (holding that resolving credibility disputes and weighing evidence was not appropriate on summary judgment).

Third, as Apple admitted early on in the district court litigation, the widgets are supplied to the accused product from third parties. Appx01595-01596 (“NXP

gives us a turn key product that has the terms that has the widget, the apps, the applet in it.”). Thus, it would not be unusual to find no direct evidence of widget code within Apple’s own source code.

3. The District Court’s Factual Resolution that Source Code Files with [REDACTED] in the File Name Could not be “Widgets” is Contradicted by the Evidence and Apple’s Own Testimony

The district court further erred in its interpretation of the source code evidence cited by Dr. Shamos. In its order, the district court cites to three excerpts from Dr. Shamos’s testimony that it claims confirm “passes are not the widget” and then infers from that interpretation that Dr. Shamos cannot imply that source code files using the word [REDACTED] may create widgets. Appx00010 (citing Dr. Shamos Deposition at 62:11-12 (Appx18963); 66:17-20 (Appx15239); and 69:20-24 (Appx18968)). But, this is an evidentiary inference not allowed on summary judgment as there is no evidence that a pass cannot be a “widget” or part of one. The district court clearly erred by assuming, if a [REDACTED] file was not itself a “widget,” then it also could not be evidence of a “widget.” Nothing about the testimony cited suggests that the use of files with [REDACTED] in their name indicates they cannot be evidence of a “widget.”

In fact, the testimony cited by the court suggests the exact opposite—that the accused product *does use* widgets. For example, at Appx18963, 62:11-12, contrary to the court’s conclusion, Dr. Shamos never indicated that “passes are not the

widget.” Instead, he testified that the file Code ID alone can’t be the widget...it can’t be all of the widget” and, Dr. Shamos went on to explain that the widget software was not necessarily one file or block of code. Appx18963-18964, 62:12-63:10. Likewise, the testimony at Appx15239, 66:17-20 never confirmed that the passes are not widgets or evidence of widgets as the court assumes. Rather, Dr. Shamos was asked if the specific Code ID file was “in the widget software that you’re accusing” and he testified that he “can’t tell” if that specific file is in the accused widget software “because...[he didn’t] have enough of it.” Appx15239. Finally, with regard to the Code ID file testimony identified at 69:20-24, Dr. Shamos explicitly testified that “[i]t is involved in the creation of the widget. I’m not ready to say that it is the widget.” Appx18968.

In other words, Dr. Shamos confirmed there were widgets, because, in addition to the evidence showing that Apple’s products use widgets, *supra* Argument Section II.A., he identified the Code ID that were at least used to create and act upon them. While such files may not themselves be the entire widget, the district court erred when it inferred from Dr. Shamos’s testimony that file names called [REDACTED] not be evidence that widgets exist because passes are not entire widgets.

Any such inference should have been construed in Fintiv’s favor—not Apple’s. *See Metro. Life Ins.*, 527 F.3d at 1338-39.

CONFIDENTIAL MATERIAL OMITTED

Indeed, Apple's own documents and witnesses indicated that passes would exhibit the same functionality as a widget that Dr. Shamos identified. In Mr. Tackin's deposition, for example, Apple indicated that passes contained card art, a user interface with software, and related to installed cards. Appx19378-19380, 87:6-14; Appx19379-19380, 166:21-167:25. Indeed, the passes appear to be terms from Apple's Passbook application that was rebranded as Apple's Wallet. Appx19329; Appx19361, 130:11-23 (describing Passbook storage and interchangeability of terminology); Appx19353 (indicating Passbook rebranded as Wallet).

Thus, the district court improperly interpreted on summary judgment to conclude passes are not evidence of widgets. To the contrary, they contain features that are synonymous with the same construction given to a "widget" by the court and that evidence cannot be summarily dismissed.

4. The District Court's Findings Regarding Other Widget Evidence were Improper and Incorrect

Next, while the district court was quick to infer how source code files that use the word **Code ID** *could not* be widgets, Appx00010, it refused to consider a similar inference in Fintiv's favor when similar code files and technical documents actually describe the code as **Code ID** Appx00008. For example, Dr. Shamos and Fintiv identified the **Code ID** file that related to **Code ID** in the product. Appx18968. However, the Court disregarded

the evidence because the file was not the entire widget. But this misses the point. Notes and comments in source code files can be used as evidence of infringement. *Versata*, 717 F.3d at 1261. Although the notes may not be the “widget” itself, they provide a reasonable inference that widgets do exist in Apple Pay during operation.

The district court further discounted the “Code ID [REDACTED] [REDACTED]” file on the grounds that it “has no purpose and is not used on the Mac” according to Apple’s employees. Appx00008. But, the court fails to address the same witness testimony that immediately follows acknowledging that the “Code ID [REDACTED]” file is not only “part of the Code ID [REDACTED] but is also “a Code ID [REDACTED] in the Code ID [REDACTED] that’s used exclusively on iOS devices, notably the iPhone.” Appx19365-19366 (emphasis added). It is a fair inference that the file at least relates to widgets in the iOS wallet application. Moreover, the only evidence to suggest it is not used comes from Apple’s own employee, but a jury can discredit that direct testimony in favor of other evidence suggesting widgets are used as previously discussed. *See Liquid Dynamics*, 449 F.3d at 1219-20.

The district court further erred in its interpretation of Dr. Shamos’s testimony that card images “‘can in fact have executable code.’” Appx00008 (quoting Fintiv brief (emphasis supplied by court)). The court dismissed the testimony as speculative because Dr. Shamos could not say whether the card image itself was

executable. But that analysis misses the point of the question Dr. Shamos was answering. Dr. Shamos was simply asked if the card image on the accused iPhone “ha[d] executable code in it,” to which he said “it can” and he explained how it could. Appx15248 (emphasis added).


However, Dr. Shamos’s testimony certainly did not speculate as to whether the accused product itself used a widget with executable code. As he testified, his testing and analysis indicated that there was a widget with executable code, *e.g.*, Appx18961; Appx18963-18964; Appx15247, 78:8-10, and he explained that while infringing code “can” reside in the PNG or PDF card image, “[i]f it doesn’t have executable code, then it is a gateway to the widget” that would have the code. Appx15247-15248.

Thus, Dr. Shamos was not speculating that there may or may not have been widget code, he was testifying that the code could be located either within the image or behind it. Appx 15247, 78:8-10. Such location disputes are not relevant to the claims, which place no requirement that executable code be within a card image file, and it is certainly not legally relevant to resolve code location disputes on summary judgment. *See Amdocs*, 761 F.3d at 1343.

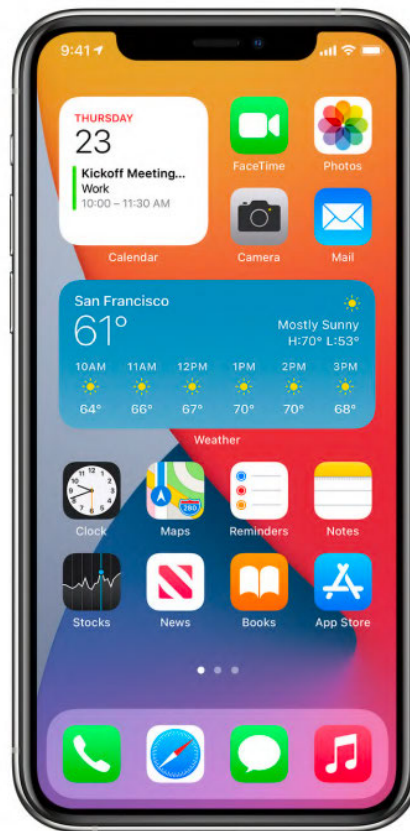
5. The District Court made Improper Findings of Fact About Related “Widget” Evidence

Finally, the district court considered product manuals and support documents that describe similar windows with informational displays and actions as “widgets.”

Add widgets to your Home Screen

1. From the Home Screen, touch and hold a widget or an empty area until the apps jiggle.
2. Tap the Add button  in the upper-left corner.
3. Select a widget, choose from three widget sizes, then tap Add Widget.
4. Tap Done.

You can also add widgets from Today View. From Today View, touch and hold a widget until the quick actions menu opens, then tap Edit Home Screen. Drag the widget to the right edge of the screen until it appears on the Home Screen, then tap Done.



Appx19307; *see also* Appx27342.

Widgets

A widget elevates key content from your app and displays it where people can see it at a glance on iPhone, iPad, and Mac. Useful and delightful, widgets can also help people personalize their iPhone Home screens in unique ways.



Appx19311; Appx27339. But, the court decided that it “finds ... those ‘other features’ have nothing to do with provisioning in Apple’s Wallet or Fintiv’s infringement claims.” Appx00008.

The district court’s finding, however, misses the point. Neither Fintiv nor its expert suggested the “other features” were part of Apple’s Wallet. Instead, Fintiv used examples of software functionality that Apple clearly admits is a “widget” to compare with similar Wallet functionality that Fintiv asserts are also “widgets.”

Indeed, the construction of “widget” is a plain and ordinary meaning of the term in the software field, Appx00063-67, and not a specialized meaning associated only with electronic payment systems or another narrow use.

Not only are Fintiv’s comparisons reasonable, they are further bolstered by the fact that Apple uses [Code ID] in its own source code descriptions of Apple’s [Code ID] Appx19365-19366; Appx18968. Thus, it is more than fair to demonstrate from pre-litigation sources that what Apple considers a [Code ID] in its [Code ID] code, corresponds to the same types of functionality it considers a “widget” in its product manuals and marketing. *See Liquid Dynamics*, 449 F.3d at 1219-20 (approving jury verdict where direct evidence indicated no infringement but court relied on product manuals and separate product installations to infer infringement by remaining products); *see also Versata*, 717 F.3d at 1263 (comparing prior art to accused product to establish infringement evidence). Making all inferences in favor of Fintiv, the fact that such “other widgets” in those manuals are similar in functionality and appearance to “widgets” in Apple’s Wallet that Dr. Shamos identified (*e.g.*, Appx18790-18791), and that Apple uses the term [Code ID] in its [Code ID] naming conventions, leads to the conclusion that Apple’s Wallet uses “widgets.”

CONFIDENTIAL MATERIAL OMITTED

III. Approving the District Court’s Demand for Source Code Sets An Extreme and Unworkable Precedent in Patent Litigation

Affirmance of the district court opinion would also result in a particularly onerous precedent that unnecessarily increases the costs and complexity of patent litigation involving software products. Although source code can be an effective way of proving infringement, it is also costly because it requires numerous hours of collection, review, and testimony by expert witnesses. *See Media Rights Techs., Inc. v. Capital One Fin. Corp.*, 800 F.3d 1366, 1374 (Fed. Cir. 2015) (acknowledging concession that court “needs expert witness testimony to determine what that source code discloses”). Also, source code is frequently considered a company’s most important trade secret. *See In re Samsung*, 2022 U.S. Dist. LEXIS 25098 at *5 (“Apple considers source code to be among its most highly confidential, proprietary, and protected business information and trade secret material. Apple expends significant time and in order to produce the source code that is the subject of Samsung’s Motion, Apple would have to expend resources and divert valuable employee time to identify the requested source code and to prepare and deploy a source code computer.” (*citing* Decl. of Apple in-house counsel)).

Indeed, most courts require extreme measures of production and review to protect it, including secure locations for viewing the code, limited ability to print source code for evidence, and restrictions upon how it can be used. *See* Appx01547-01556 (Agreed Protective Order in this matter setting limits on location of code,

viewing of code, and printing of code excerpts); *see also Drone Techs., Inc. v. Parrot S.A.*, 838 F.3d 1283, 1300 n.13 (Fed. Cir. 2016) (“[I]t is well recognized among lower courts that source code requires additional protections to prevent improper disclosure because it is often a company's most sensitive and most valuable property.”); *Via Vadis Controlling GmbH v. Skype, Inc.*, Civil Action No. 12-mc-193-RGA, 2013 U.S. Dist. LEXIS 23434, at *7 (D. Del. Feb. 21, 2013) (recognizing in response to request for production of source code that a “general request for the source code and related documents places a heavy burden on Respondents. Source codes are the most sensitive and confidential property of Respondents. When disclosed in U.S. litigation, extreme measures are ordered to protect their confidentiality.”). Further still, evidence of some product functionality may be held in third party source code that is not a party to the case. *See Advanced Micro Devices, Inc. v. LG Elecs., Inc.*, 14-cv-01012-SI, 2017 U.S. Dist. LEXIS 110776, at *8 (N.D. Cal. Jul. 17, 2017) (requiring disclosure of third party, Imagination Technologies, LLC, source code).

As a result of how burdensome source code production and review can be, parties often attempt to limit source code production and review to only code that is strictly necessary. *See Laserdynamics, Inc. v. Asus Computer, Int’l et al.*, 2:06-cv-348, 2009 U.S. Dist. LEXIS 3878, *7-9, 13, 15 (E.D. Tex. Jan. 21, 2009) (ordering third party to disclose source code with specificity). Thus, rather than spend

increasing amounts of money on experts to review thousands of pages of code across multiple versions of an accused product, parties will seek to prove software functionality without resorting to source code for every claim element, *e.g.*, through deposition testimony, product manuals, product testing, etc. While such means may not be direct evidence of what the actual source code says, the decision to rely upon circumstantial evidence of how a device operates can still be persuasive to the ultimate fact finder and indeed may be more persuasive. *See Versata*, 717 F.3d at 1261-62; *Liquid Dynamics*, 449 F.3d at 1219-20, 1226 (upholding that jury's finding for infringement based on expert simulations and demonstration of engineering manuals).

Indeed, concerns over source code protection were even articulated multiple times in this case as Apple fought to protect not only the source code, but also the file name conventions under the strictest confidentiality standards. Appx03781-03784 (denying request by Apple to treat file names identified in contentions under strict source code protections and replace such names in contentions). But Apple, and parties like it, should not be allowed to use source code restrictions as a sword and shield, in which it shields disclosure of code on the basis of costs and confidentiality protections, but then argues a patent owner cannot prove infringement because it did not present every element of a claim with direct source code evidence. The fact that a patent owner chooses to forego the cumbersome and

highly confidential nature of source code review for each and every element of a claim when circumstantial evidence should suffice does not warrant automatic summary judgment against them. If anything, it should be acknowledged as a way to reduce and not increase the judicial burden of handling such sensitive and confidential information.

The district court's willingness to grant summary judgment after a disproportionate focus on the lack of source code, to the exclusion of other expert and circumstantial evidence, would virtually require parties to collect, review, produce, and present as evidence more and more source code from each other and third parties even though such code may not be strictly necessary to prove certain elements. While source code may be evidence of infringement by software, it should not be the only—or even the preferred—form of evidence by courts, or else such a preference will virtually mandate discovery and protection of large volumes of code from parties and non-parties to a litigation in order to avoid summary judgment.

Furthermore, source code details are often strictly maintained by defendants in a patent infringement suit and are only made available during discovery. Thus, any requirement or preference for source code would place possibly the only acceptable evidence of infringement under the sole control of a defendant regardless of what can be demonstrated by experts about the functionality of the product. Such discretion is exceedingly worrisome.

CONCLUSION

The case should be reversed and remanded because the district court improperly required source code as evidence and failed to consider other relevant evidence of a widget in Apples' accused devices.

November 16, 2023

Respectfully submitted,

Meredith Martin Addy

Jonathan K. Waldrop
Darcy L. Jones
Marcus A. Barber
John W. Downing
Heather S. Kim
ThucMinh Nguyen
Kasowitz Benson Torres LLP
333 Twin Dolphin Drive
Suite 200
Redwood Shores, CA 94065
650.453.5170
jwaldrop@kasowitz.com
djones@kasowitz.com
mbarber@kasowitz.com
jdowning@kasowitz.com
hkim@kasowitz.com
tnguyen@kasowitz.com

Meredith Martin Addy
Charles A. Pannell, III
ADDYHART P.C.
10 Glenlake Parkway, Suite 130
Atlanta, Georgia 30328
312.320.4200
meredith@addyhart.com
cpannell@addyhart.com

Caren A. Yusem
ADDYHART P.C.
1101 Pennsylvania Avenue, N.W.
Suite 300
Washington, DC 20004
312.804.4885
caren@addyhart.com

Paul G. Williams
Kasowitz Benson Torres LLP
1230 Peachtree Street, NE
Suite 2445
Atlanta, GA 30309
404.260.6102
pwilliams@kasowitz.com

**CERTIFICATE OF COMPLIANCE WITH
TYPE-VOLUME LIMITATIONS**

Fintiv, Inc. v. Apple Inc.

23-2208

This brief complies with the relevant type-volume limitation of the Federal Rule of Appellate Procedure and Federal Circuit Rules because it has been prepared using a proportionally-spaced typeface and includes 12,485 words.

Dated: November 16, 2023

By: /s/ *Meredith Martin Addy*

Meredith Martin Addy

ADDENDUM

District Court Final JudgmentAppx00001

District Court Order Granting Motion for Summary Judgment.....Appx00002

U.S. Patent No. 8,843,125 B2Appx00085

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION**

FINTIV,

Plaintiff,

v.

APPLE INC.,

Defendant.

Civil Action No. 1:21-cv-00896

FINAL JUDGMENT

The Court granted Defendant Apple Inc.'s Motion for Summary Judgment of No Infringement on June 21, 2023. ECF No. 467. The Court now enters its Judgment as follows:

IT IS ORDERED that final judgment is entered in favor of Defendant and against Plaintiff. Plaintiff shall take nothing by this action.

IT IS FURTHER ORDERED that any and all motions not previously ruled upon by the Court are **DENIED** as moot.

IT IS FURTHER ORDERED that Defendant's remaining counterclaims and defenses are dismissed without prejudice.

IT IS FURTHER ORDERED that any relief not specifically granted in this judgment is **DENIED**.

IT IS FINALLY ORDERED that the Clerk of Court is directed to close the case.

SIGNED this 29th day of June, 2023.



ALAN D ALBRIGHT
UNITED STATES DISTRICT JUDGE

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION**

FINTIV, INC.,
Plaintiff,

v.

APPLE INC.,
Defendant.

1:21-CV-896-ADA

**ORDER GRANTING MOTION
FOR SUMMARY JUDGMENT [ECF No. 270]**

Before the Court is Defendant Apple Inc.’s (“Apple”) Motion for Summary Judgment of Non-Infringement. ECF No. 270. Plaintiff Fintiv, Inc. (“Fintiv”) filed a Response. ECF No. 300. Apple then replied. ECF No. 348. After originally denying the motion, the Court heard further oral argument on the Motion on June 13, 2023. ECF Nos. 465–66. At the hearing, the Court granted Apple’s Motion and vacated its prior decision. This opinion memorializes that ruling.

I. BACKGROUND

On December 21, 2018, Fintiv filed its complaint alleging infringement of U.S. Patent No. 8,843,125 (“the ’125 patent”). ECF No. 1 ¶ 3. Fintiv alleges Apple infringes independent claims 11, 18, and 23 and dependent claims 13, 14, 20, 24, and 25 (“asserted claims”). ECF No. 270-2 (“Shamos Report”) ¶ 2. All the asserted claims relate generally to “card provisioning.” ECF No. 270-3 (“Shamos Depo.”) at 31:21–24; ECF No. 273-1 (“’125 patent”) claims 11, 18, 23. Card provisioning is a process whereby a user “load[s] data concerning a payment instrument, such as a credit card, onto a mobile device for the purposes of making payment transactions.” Shamos Report ¶ 71. Independent claim 11 recites a method for card provisioning, specifically a “method for provisioning a contactless card applet in a mobile device comprising a mobile wallet application.” Independent claim 18 recites a system for card provisioning, specifically a “wallet

management system (WMS) in a non-transitory storage medium to store and manage mobile wallet account information.” Independent claim 23 recites a “mobile device” for card provisioning. ’125 patent, claims 11, 18, 23.

Fintiv accuses each of the Apple iPhone, Watch, iPad, and Mac products of infringing at least one claim of the ’125 patent. Shamos Report ¶¶ 102–03. Every asserted claim recites a “widget.” Claim 11 requires “retrieving a widget . . . corresponding to a contactless card applet” and “provisioning the widget.” Claim 18 requires “a widget management component configured to store and to manage widgets” and “a rule engine configured to filter a widget.” Claim 23 requires “a mobile wallet application configured to store a widget” and “an over-the-air (OTA) proxy configured to provision . . . a widget.” The Court construed “widget” to have its plain-and-ordinary meaning, where the plain-and-ordinary meaning is “software that is either an application or works with an application, and which may have a user interface.” ECF No. 86 at 17, 34. The Court also ruled that “a POSITA would not understand that a widget is a stand-alone application, but rather as code, e.g., a ‘plug-in,’ that runs within an application.” *Id.* at 16.

II. LEGAL STANDARD

Summary judgment is appropriate “if the movant shows that there is no genuine dispute as to any material fact and the movant is entitled to judgment as a matter of law.” FED. R. CIV. P. 56(a); *Tolan v. Cotton*, 572 U.S. 650, 656–57 (2014). A material fact will have a reasonable likelihood to affect the outcome of the case. *Anderson v. Liberty Lobby, Inc.*, 477 U.S. 242, 248 (1986). An issue is not genuine if the trier of fact could not, after an examination of the record, rationally find for the non-moving party. *Matsushita Elec. Indus., Co. v. Zenith Radio Corp.*, 475 U.S. 574, 587 (1986). As such, the burden of demonstrating a lack of a genuine dispute of material fact lies with the movant. *Celotex Corp. v. Catrett*, 477 U.S. 317, 323 (1986).

A court must view the movant's evidence and all factual inferences from such evidence in a light most favorable to the party opposing summary judgment. *Impossible Elecs. Techniques v. Wackenhut Protective Sys., Inc.*, 669 F.2d 1026, 1031 (5th Cir. 1982). Accordingly, the fact that the court believes that the non-moving party will be unsuccessful at trial is an insufficient reason to grant summary judgment in favor of the moving party. *See Jones v. Geophysical Co.*, 669 F.2d 280, 283 (5th Cir. 1982). Yet, "[w]hen opposing parties tell two different stories, but one of which is blatantly contradicted by the record, so that no reasonable jury could believe it, a court should not adopt that version of the facts for the purposes of ruling on a motion for summary judgment." *Scott v. Harris*, 550 U.S. 372, 380–81 (2007).

Once the court determines that the movant has presented sufficient evidence that no genuine dispute of material fact exists, the burden of production shifts to the party opposing summary judgment. *Matsushita*, 475 U.S. at 586. The non-moving party must demonstrate a genuinely disputed fact by citing to parts of materials in the record, such as affidavits, declarations, stipulations, admissions, interrogatory answers, or other materials; or by showing that the materials cited by the movant do not establish the absence of a genuine dispute. FED. R. CIV. P. 56(c)(1)(A)–(B). "Conclusory allegations unsupported by concrete and particular facts will not prevent an award of summary judgment." *Duffy v. Leading Edge Prods.*, 44 F.3d 308, 312 (5th Cir. 1995).

III. DISCUSSION

Apple moves for summary judgment of non-infringement of the '125 patent on four independent grounds. One of Apple's grounds is that Fintiv and its technical expert, Dr. Michael Shamos, identified no software in the accused products that constitutes a "widget" under the Court's construction of that term—a requirement of all asserted claims. As explained below, the Court finds that the record is devoid of evidence that the accused products practice the "widget" limitation under the Court's construction. Because the Court finds that Apple has met its burden

on this ground, the Court need not address the other three grounds. Apple has therefore established that there is no genuine dispute of material fact that it does not practice the asserted claims, and it is entitled to summary judgment as a matter of law.

A. Apple's Position

Apple contends that it is entitled to summary judgment of noninfringement as to all asserted claims because (1) Fintiv identified no software code in the accused products that meets the “widget” limitations of the asserted claims, and (2) undisputed facts confirm the accused products do not use and are not configured to use a “widget.” ECF No. 270 at 12–13. *First*, Apple argues that Fintiv’s expert Dr. Shamos identified no Apple software code that constitutes a “widget” as construed by the Court. *Id.* at 14. Indeed, Apple points to Dr. Shamos’s deposition testimony that confirms this argument:

Q. So do we agree that your report does not cite the -- or identify the software that is the widget in the accused iPhone device?

A. Yeah. I think it -- I think it doesn't identify the source code of the widget.

Id. (citing Shamos Depo. at 73:12–74:5 (emphasis added)). After citing several occasions in Dr. Shamos’s deposition testimony establishing there is no source code that makes up the widget in the accused devices, Apple reiterates that Dr. Shamos confirmed under oath that “there is nowhere in [his] report that cites the source code that makes up the widget for any of the accused devices.” *See id.* at 15 (citing Shamos Depo. at 75:14–20).

Second, Apple asserts that Fintiv also failed to present evidence that the accused products practice other “widget”-related limitations of the asserted claims. *Id.* For example, asserted claim 11 requires “retrieving a widget.” Dr. Shamos agreed that during card provisioning a widget must be retrieved from an Apple server to an accused device, but Apple argues he could not identify any widget retrieved from any Apple server. *Id.* (citing Shamos Depo. at 56:23–25). And asserted claim 18 requires “a widget management component configured to store and to manage widgets,”

but Apple contends that Dr. Shamos could not identify which server allegedly stores the claimed “widget,” nor does he know which component is responsible for such storage. *Id.* at 16.

B. Fintiv And Its Expert Failed To Identify The Claimed “Widget” In The Accused Products.

Fintiv and Dr. Shamos failed to identify the claimed widget in the accused products. In Dr. Shamos’ expert report, only one paragraph discusses the “widget” limitation and contains citations to Apple’s source code—paragraph 309. Shamos Report ¶ 309. But when asked at his deposition, Dr. Shamos conceded that none of the source code cited in that paragraph is the claimed “widget”:

Q. So the software that you’re talking about in Paragraph 309 -- that is the widget, is that the software that’s cited in Paragraph 309?

A. I don’t think so.

Shamos Depo. at 53:11–14. When Dr. Shamos was asked about each of the source code files cited in paragraph 309 individually, he confirmed that none of those files is a “widget.” *Id.* at 61:21–23 (██████████ not the widget); 62:11–12 (“██████████ alone can’t be the widget”); 66:17–20 (“I can’t tell” if ██████████ is the widget); 69:20–24 (“not ready to say” ██████████ is the widget); 70:11–12 (██████████ not the widget); 70:16–22 (██████████ and ██████████ not the widget); 71:14–17 (██████████ not the widget); 71:18–24 (██████████ not the widget); 71:25–72:5 ██████████ ██████████ not the widget). As explained above, Dr. Shamos also confirmed that none of the software files cited in other parts of his report constitutes a “widget”:

Q. But if we did the same exercise we just spent the last 30 minutes doing, we would find that there is nowhere in your report that cites the source code that makes up the widget for any of the accused devices. Is that right?

A. That’s right.

Id. at 75:14–20 (emphasis added).

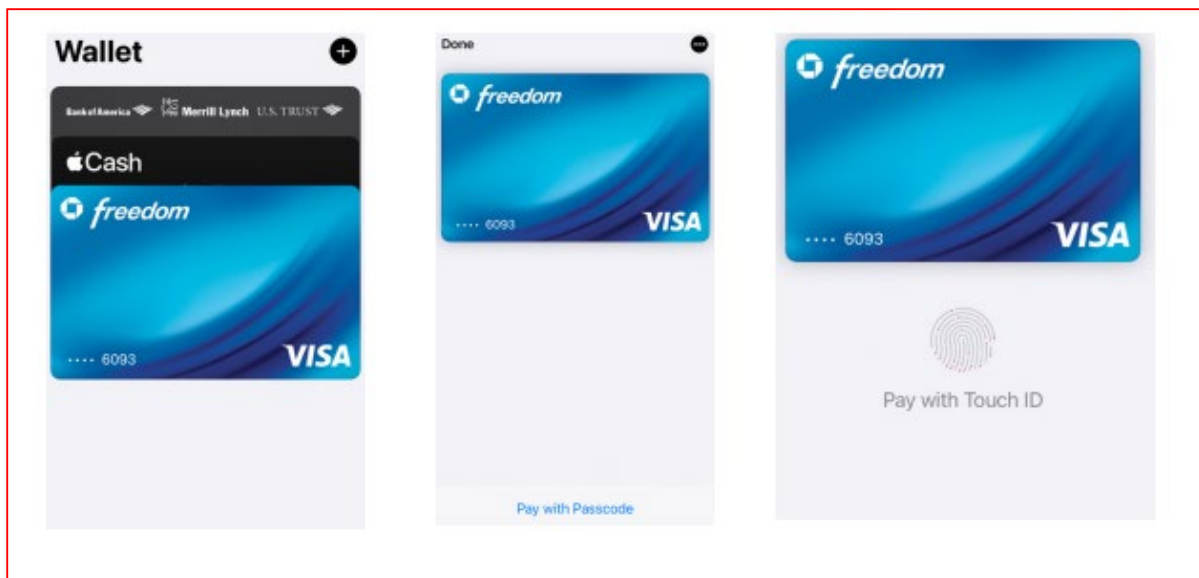
Fintiv’s opposition confirms it has no evidence of a “widget” in the accused products; that is, the accused products contain no “software that is either an application or works with an application, and which may have a user interface.” *See* ECF No. 86 at 17, 34. Fintiv first responds that Apple is misreading the construction of “widget” to require “software code” because the word “code” is not the Court’s construction of “widget” and source code is not the only way Fintiv can prove Apple’s infringement. *See* ECF No. 300 at 9–10. Not so. Fintiv cites only two cases in support of its arguments (*see id.* at 10), but neither case applies here. In *Tarkus Imaging Inc. v. Adobe Sys.*, Case No. 10-63-LPS, 2012 WL 2175788 (D. Del. June 14, 2012), the court denied summary judgment of noninfringement because the court was “not persuaded” by the fact that “Tarkus’s expert has not identified specifically infringing source code.” *Id.* at *3. But unlike Dr. Shamos, the plaintiff’s expert in *Tarkus* relied on “testimony from a Canon Rule 30(b)(6) witness on source code [and] the expert report of William Elswick, an expert for Tarkus who personally reviewed the source code.” *Id.* at *3 n.2. Here, Dr. Shamos cites no deposition testimony nor any expert who “personally reviewed” the Apple source code to support his speculation that there must be a “widget” in the products.¹ Critically, Apple is not misconstruing the court’s construction of “widget” to require source code; Apple is showing the complete devoid of evidence that Dr. Shamos presented in his expert report to prove infringement.

¹ *i4i Ltd. Partnership v. Microsoft Corp.*, 598 F.3d 831, 847–48 (Fed. Cir. 2010), also does not support Fintiv. Contrary to Fintiv’s misleading parenthetical (“affirming jury finding of infringement despite no source code available to be presented,” ECF No. 300 at 10), the lack of source code related to invalidity, not infringement. The prior art source code had been destroyed years before litigation began, so the dispute “turned largely on the credibility of [the prior art’s] creators.” *Id.* at 846. The court affirmed the jury’s finding of validity because “the jury was free to disbelieve Microsoft’s expert ... and credit i4i’s expert, who opined that it was impossible to know whether the claim limitation was met without looking at S4’s source code.” *Id.* at 848 (emphasis added). Here, Dr. Shamos had Apple’s source code, but still found no “widget.”

Fintiv then contends that it has proffered sufficient evidence (source code and non-source code) to defeat Apple’s Motion. *See* ECF No. 300 at 9. But Fintiv’s opposition never identifies what “software” or “software code” comprises the claimed “widget” in the accused products. *See* ECF No. 300 at 9–23. Fintiv instead points to multiple things that are not the claimed “widget.” For instance, Fintiv claims that “Apple’s own technical documents reference ‘widgets’ and ‘widget’ in connection with *other features*.” *Id.* at 11–12 (emphasis added, pointing to Home Screen widgets); *see also id.* at 16 (“Apple has marketed other features as ‘widgets.’”). The Court finds that those “other features” have nothing to do with provisioning in Apple Wallet or Fintiv’s infringement claims. Fintiv also claims that Apple servers allegedly “build widget assets,” but Fintiv does not say what those “assets” purportedly are. *Id.* at 13. Fintiv also speculates that an “underlying file that produces the image of a card on the screen can in fact have executable code” (*id.* at 21, emphasis added), but speculation is not a substitute for evidence. And Fintiv points to a source code module, [REDACTED] (*id.* at 18), that “has no purpose and is not used on the Mac,” but never says that module is the claimed “widget”—because it is not. ECF No. 270-6 (“Diederich Depo.”) at 18:4–7; 114:17–25.

Fintiv also points to Dr. Shamos’s report at paragraphs 359–60. ECF No. 300 at 13. There, Dr. Shamos relies on the following three screenshots to show that “the widget (providing a user interface) is also provisioned (made available for use).” *Id.*

CONFIDENTIAL MATERIAL REDACTED.



Shamos Report ¶¶ 359–60. But neither Dr. Shamos nor Fintiv can identify specifically what in these screenshots is the claimed “widget.” *See id.* Dr. Shamos merely states that “[e]ach screenshot below (showing the virtual card image for the Visa card) presents a software (with a user interface) that is made available to the user for selecting, via its user interface, among the available ones to, for example, view the card's details or perform transactions.” *Id.* ¶ 359. This is mere speculation and is insufficient to establish a genuine dispute of material fact that the accused products infringe the “widget” limitation.

Fintiv’s opposition relies on source code modules that its own expert testified under oath are not the claimed “widget.” *See* ECF No. 300 at 15 (“The foregoing source code modules of Apple’s servers are used to create widgets that are stored in Apple’s servers.”). Dr. Shamos was asked about each of the cited source code modules, and he confirmed under oath that none of them is the accused “widget.” *See* Shamos Depo. at 61:21–23 ([REDACTED] is not the widget); 70:11–12 ([REDACTED] is not the widget); *see also id.* at 53:11–14; 62:11–12; 66:17–20; 69:20–24; 70:16–22; 71:14–17; 71:18–24; 71:25–72:5 (each of the source code files

cited in Shamos Report ¶ 309 is not the claimed widget). As the file names reveal, Fintiv is pointing to passes, implying they might be “created” as the “widget.” *See* ECF No. 300 at 15, 18. But Dr. Shamos also confirmed that no matter what creates them, passes are not the widget. Shamos Depo. at 62:11–12; 66:17–20; 69:20–24. Thus, nowhere does Fintiv’s opposition state that “the ‘widget’ in the accused product is X,” where X is an identifiable piece of software, as required by the Court’s construction.

C. Dr. Shamos’ Speculation About Nonexistent “Widgets” In The Accused Products Is Not Enough To Survive Summary Judgment.

Faced with its failure to identify software that constitutes the accused “widget,” Fintiv cites Dr. Shamos’ speculation that there must be a widget somewhere in the accused products, even though he failed to identify it in his expert report. *Compare* ECF No. 300 at 9 (“And--are you prepared to testify at trial that there is a widget in the accused Apple devices that infringes the claims of the ’125 patent? A. Yes.”) *with* Shamos Depo. at 73:12–74:5 (agreeing that his report fails to identify the source code or software that is the accused widget). But under settled Federal Circuit law, “the non-movant can’t defeat summary judgment with conclusory allegations, unsupported assertions, or only a scintilla of evidence.” *Traxcell Techs., LLC v. Sprint Commc’ns Co. LP*, 15 F.4th 1121, 1128 (Fed. Cir. 2021) (citing *Batiste v. Lewis*, 976 F.3d 493, 500 (5th Cir. 2020)). Rather, a plaintiff must prove with admissible evidence that the accused products meet each limitation of an asserted claim. *See Catalina Mktg. Int’l, Inc. v. Coolsavings.com, Inc.*, 289 F.3d 801, 812 (Fed. Cir. 2002). Thus, Fintiv cannot survive summary judgment by citing testimony that its expert is prepared to speculate at trial “that there is a widget in the accused Apple devices.” ECF No. 300 at 9. Simply saying, “it must be in there somewhere” is no substitute for the requirement that Fintiv “set forth specific facts showing that there is a genuine issue for trial” with respect to the “widget” limitation. *See Anderson*, 477 U.S. at

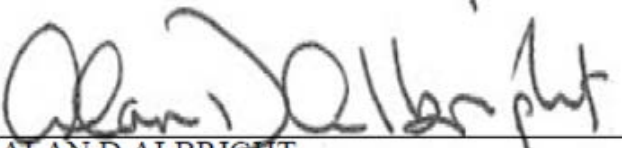
256. Fintiv’s opposition is devoid of any such “specific facts,” because it cannot identify a “widget” in the accused card provisioning process.

On the contrary, Fintiv doubles down on its speculation by arguing that “Apple’s own technical documents reference ‘widgets’ and ‘widget’ in connection with other features.” ECF No. 300 at 11 (emphasis added), citing ECF No. 300 Exs. 13, 14. As Fintiv admits, the cited evidence has nothing to do with Apple Pay or Apple Wallet; rather, the evidence relates to “us[ing] widgets on your Home Screen.” *Id.* And not surprisingly, Dr. Shamos cites none of this evidence in his expert report, because it does not bear on infringement.² So Fintiv’s speculation—without even the support of its expert—that “there must be widgets in Apple Pay because there are widgets in other parts of the product” is even further removed from the “specific facts” Fintiv was required to set forth in opposing the Motion. *Anderson*, 477 U.S. at 256.

IV. CONCLUSION

Apple has presented sufficient evidence that no genuine dispute of material fact exists on non-infringement. Fintiv then failed to demonstrate a genuinely disputed material fact, or indeed any facts at all, showing that Apple infringes the “widget” limitation present in all asserted claims of the ’125 patent. Accordingly, Defendant Apple’s Motion for Summary Judgment of Non-Infringement (ECF No. 270) is **GRANTED**.

SIGNED this 21st day of June, 2023.


ALAN D ALBRIGHT
UNITED STATES DISTRICT JUDGE

² Fintiv also cites an undated, one-page presentation slide titled “Provisioning: Requirements.” ECF No. 300 at 13. Dr. Shamos neither cites nor discusses this document, and it says nothing about “software that is an application or works with an application.” The document refers to data that is sent and received, including “required fields for the card to be provisioned,” “payment product name,” and whether “card supported or not.” The Court finds this document is not evidence of the claimed “widget.”



US008843125B2

(12) **United States Patent**
Kwon et al.

(10) **Patent No.:** **US 8,843,125 B2**
(45) **Date of Patent:** **Sep. 23, 2014**

(54) **SYSTEM AND METHOD FOR MANAGING MOBILE WALLET AND ITS RELATED CREDENTIALS**

(75) Inventors: **Yongsung Kwon**, Seongnam-si (KR); **Hyungjoon Hong**, Seoul (KR); **Jiwon Kang**, Seoul (KR); **Hyunjin Kim**, Yongin-si (KR)

(73) Assignee: **SK C&C**, Seongnam, Gyeonggi-Do (KR)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 446 days.

(21) Appl. No.: **13/310,091**

(22) Filed: **Dec. 2, 2011**

(65) **Prior Publication Data**

US 2012/0172026 A1 Jul. 5, 2012

Related U.S. Application Data

(60) Provisional application No. 61/428,846, filed on Dec. 30, 2010, provisional application No. 61/428,851, filed on Dec. 30, 2010, provisional application No. 61/428,852, filed on Dec. 30, 2010, provisional application No. 61/428,853, filed on Dec. 30, 2010.

(51) **Int. Cl.**

H04W 4/00 (2009.01)
H04W 12/04 (2009.01)
H04L 29/06 (2006.01)
H04W 12/06 (2009.01)

(52) **U.S. Cl.**

CPC **H04W 12/06** (2013.01); **H04W 12/04** (2013.01); **H04L 63/067** (2013.01)
USPC **455/419**; **455/410**

(58) **Field of Classification Search**

USPC 455/410, 418, 419, 558; 705/16, 39, 41
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,221,838	A	6/1993	Gutman et al.
6,199,762	B1	3/2001	Hohle
6,480,957	B1	11/2002	Liao et al.
6,487,403	B2	11/2002	Carroll
6,950,939	B2	9/2005	Tobin
7,024,390	B1	4/2006	Mori et al.
7,065,341	B2	6/2006	Kamiyama et al.
7,146,159	B1	12/2006	Zhu
7,149,545	B2	12/2006	Hurst et al.
7,155,411	B1	12/2006	Blinn et al.
7,197,297	B2	3/2007	Myles et al.

(Continued)

OTHER PUBLICATIONS

GlobalPlatform, Card Specification, Version 2.2, published Mar. 2006.

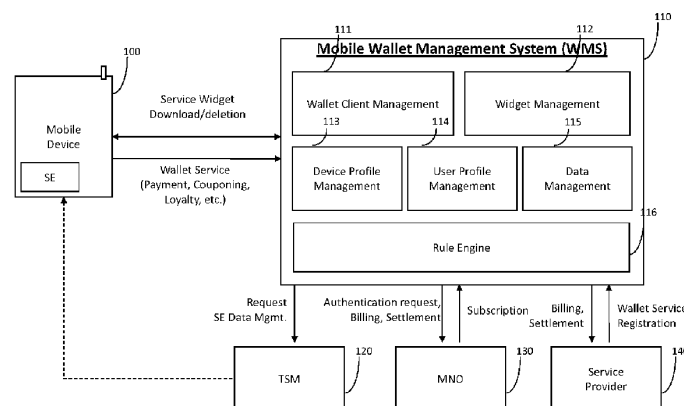
Primary Examiner — Sam Bhattacharya

(74) *Attorney, Agent, or Firm* — Lowe Hauptman & Ham, LLP

(57) **ABSTRACT**

A method for provisioning a contactless card applet in a mobile device with a mobile wallet application, including activating the mobile wallet application, connecting to a Trusted Service Manager (TSM) system, synchronizing the mobile wallet application with the TSM system, displaying a contactless card applet based on attributes of the mobile device, receiving a selection of a contactless card applet, retrieving a widget and a wallet management applet (WMA) corresponding to the contactless card applet, and provisioning the selected contactless card applet, the widget, and the WMA. A wallet management system (WMS) in a non-transitory storage medium to store and manage mobile wallet account information including a wallet client management component, a widget management component, a device profile management component, a user profile management component, a data management component, a rule engine, a subscription management component, a billing management component, a settlement management component, and a rule engine.

25 Claims, 5 Drawing Sheets



US 8,843,125 B2

Page 2

(56)

References Cited

U.S. PATENT DOCUMENTS

7,233,785 B2	6/2007	Yamagishi et al.	7,822,439 B2	10/2010	Teicher
7,233,926 B2	6/2007	Durand et al.	7,822,688 B2	10/2010	Labrou et al.
7,236,742 B2	6/2007	Hall et al.	2008/0010215 A1	1/2008	Rackley III et al.
7,286,818 B2	10/2007	Rosenberg	2008/0040265 A1	2/2008	Rackley III et al.
7,389,123 B2	6/2008	Rydgren et al.	2008/0208742 A1	8/2008	Arthur et al.
7,415,721 B2	8/2008	Fransdonk	2009/0124234 A1	5/2009	Fisher et al.
7,447,494 B2	11/2008	Law et al.	2009/0307139 A1	12/2009	Mardikar et al.
7,454,233 B2	11/2008	Lu et al.	2009/0307140 A1	12/2009	Mardikar
7,469,151 B2	12/2008	Khan et al.	2010/0125495 A1	5/2010	Smith et al.
7,490,775 B2	2/2009	Biderman	2010/0125508 A1	5/2010	Smith
7,527,208 B2	5/2009	Hammad et al.	2010/0138518 A1	6/2010	Aiglstorfer et al.
7,628,322 B2	12/2009	Holtmanns et al.	2010/0145835 A1	6/2010	Davis et al.
7,689,205 B2	3/2010	Toy et al.	2010/0205432 A1	8/2010	Corde et al.
7,689,508 B2	3/2010	Davis et al.	2010/0211507 A1	8/2010	Aabye et al.
7,707,113 B1	4/2010	DiMartino et al.	2010/0275242 A1	10/2010	Raffard et al.
7,708,194 B2	5/2010	Vawter	2010/0275269 A1	10/2010	Vilmos et al.
7,711,392 B2	5/2010	Brown et al.	2010/0291904 A1	11/2010	Musfeldt et al.
7,819,307 B2	10/2010	Lyons et al.	2010/0306107 A1	12/2010	Nahari
			2010/0330958 A1	12/2010	Corde et al.
			2011/0078081 A1	3/2011	Pirzadeh et al.
			2014/0089185 A1 *	3/2014	Desai et al. 705/41

* cited by examiner

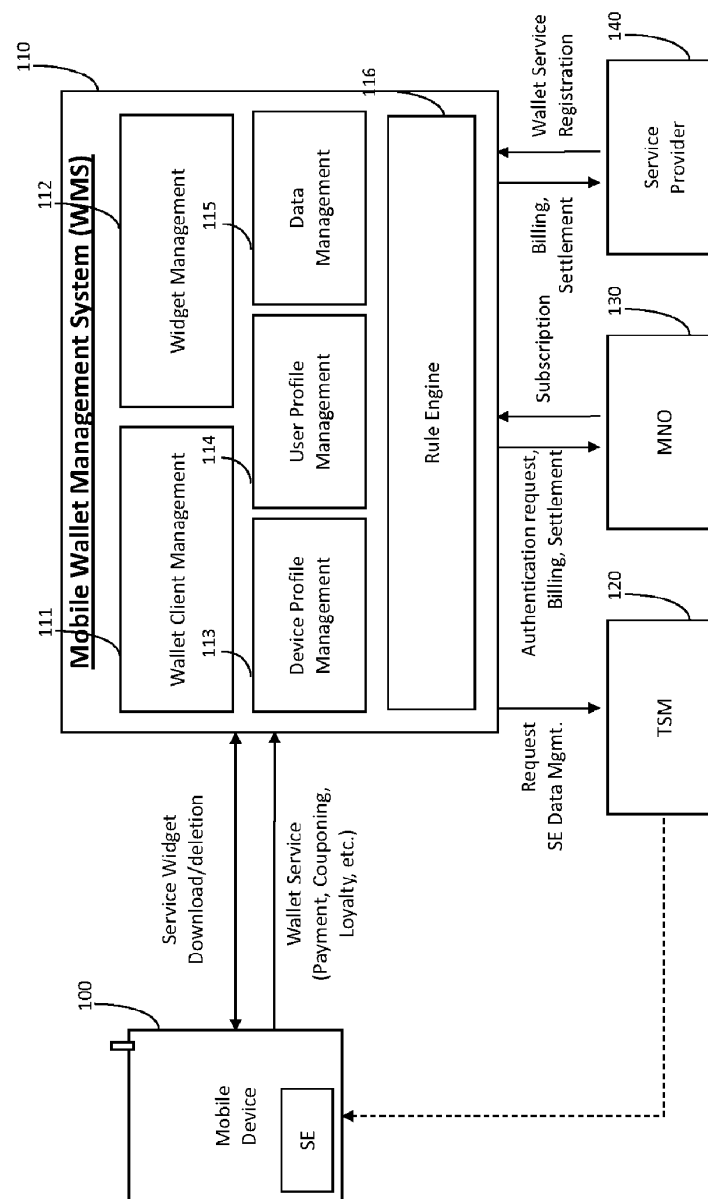
U.S. Patent

Sep. 23, 2014

Sheet 1 of 5

US 8,843,125 B2

Fig. 1



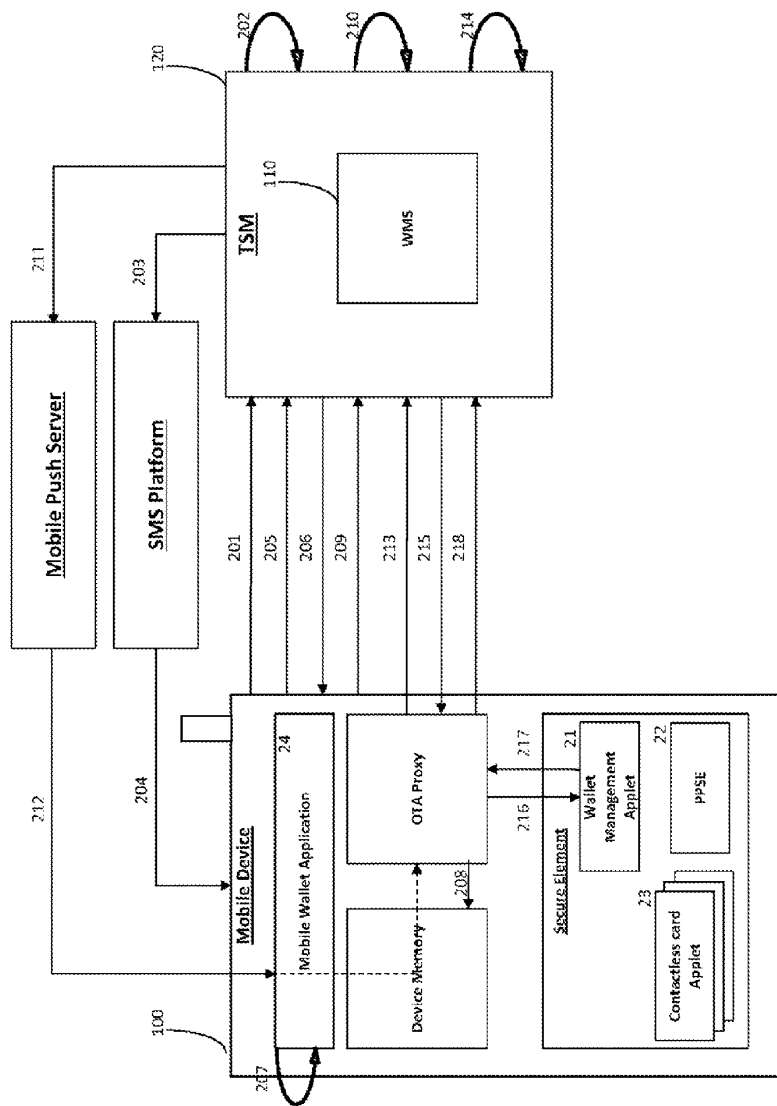
U.S. Patent

Sep. 23, 2014

Sheet 2 of 5

US 8,843,125 B2

Fig. 2. Install Wallet Application



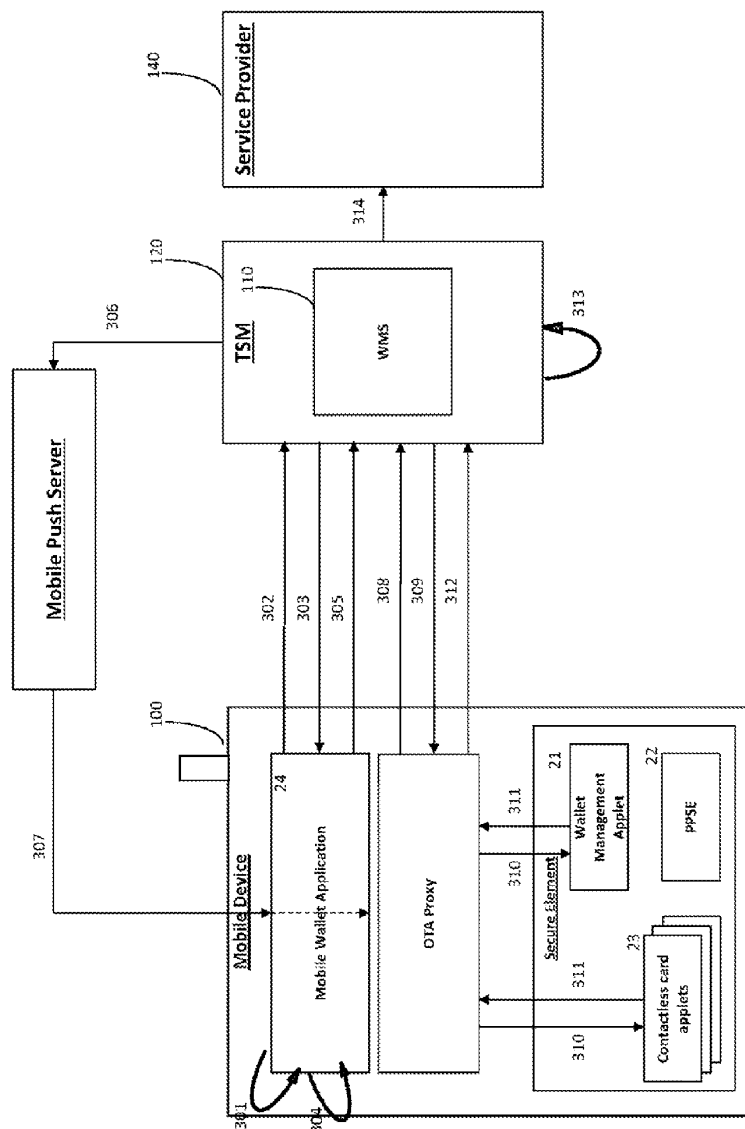
U.S. Patent

Sep. 23, 2014

Sheet 3 of 5

US 8,843,125 B2

Fig. 3. Install Widget (prior SP reg)

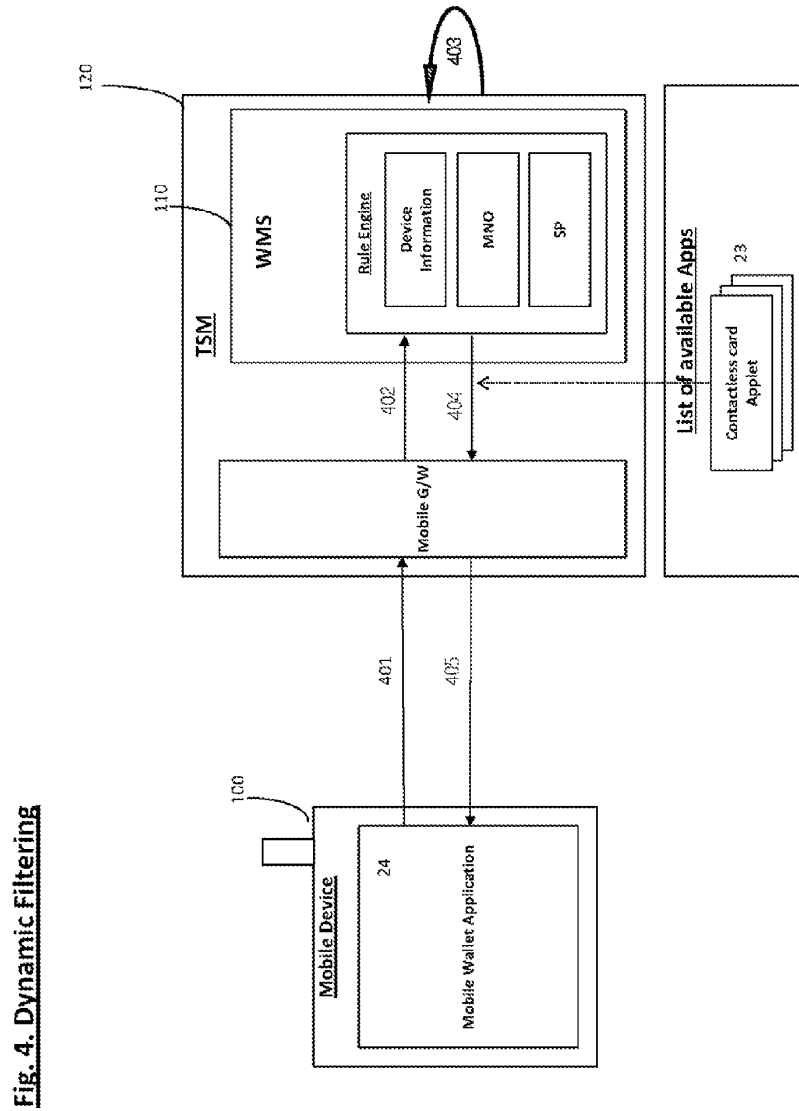


U.S. Patent

Sep. 23, 2014

Sheet 4 of 5

US 8,843,125 B2



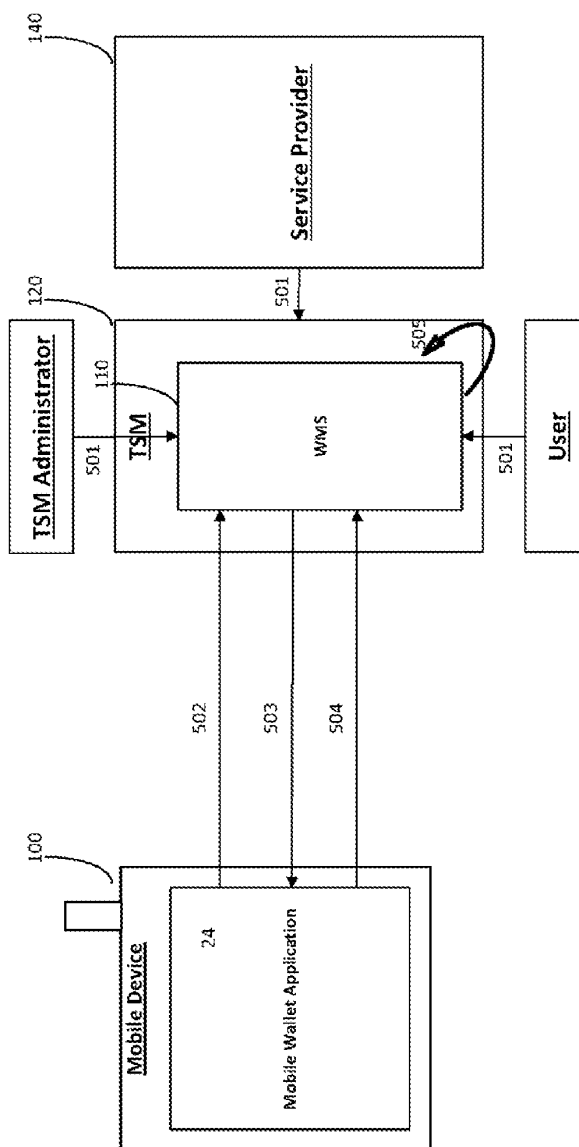
U.S. Patent

Sep. 23, 2014

Sheet 5 of 5

US 8,843,125 B2

Fig. 5. Synchronization



US 8,843,125 B2

1

SYSTEM AND METHOD FOR MANAGING MOBILE WALLET AND ITS RELATED CREDENTIALS

CROSS REFERENCE TO RELATED APPLICATION

This application claims priority from and the benefit under 35 U.S.C. §119(a) of U.S. Provisional Patent Application No. 61/428,846, filed on Dec. 30, 2010, which is incorporated by reference for all purposes as if fully set forth herein. Also, the present application is related to U.S. Provisional Patent Application No. 61/428,851 filed on Dec. 30, 2010; U.S. Provisional Patent Application No. 61/428,852, filed on December 30; and U.S. Provisional Patent Application No. 61/428,853, filed on December 30. Applicant hereby incorporates by reference the above-mentioned provisional applications, which are not admitted to be prior art with respect to the present invention by their mention here or in the background section that follows.

BACKGROUND OF THE INVENTION

1. Field

The following description relates to management of virtual cards stored on mobile devices.

2. Discussion of the Background

With the advent of advancing mobile technology, more features have been integrated into mobile devices. From GPS applications to mobile office products, mobile devices, such as mobile communicative terminals, have practically become a necessity for everyday needs. In order to further utilize mobile technology to better cater to a user's daily requirements, attempts have been made to provide for a mobile financial management system to replace conventional physical wallets. Specifically, this mobile wallet functionality was sought to be realized through provisioning of card issuer's account information directly into a secure element (SE) of the mobile device equipped with Near Field Communication (NFC) chipset. The SE may be a smart card chip capable of storing multiple applications, including of account specific information that may not be easily accessed by external parties. The model mobile wallet application may have the same composition as a conventional wallet, which may contain payment cards, member cards, transportation cards, and loyalty cards.

Further, to make the wallet function more convenient to the owners of the mobile device, a method of providing contactless payment (NFC-based applications) through provisioning account specific information within the secure domain of the mobile device's SE has been provided. More specifically, user financial credentials, such as credit card numbers, may be provisioned onto mobile devices equipped with Near Field Communication chipset (NFC enabled) to make payments. Once the user financial credentials have been provisioned onto the NFC enabled mobile device, the provisioned NFC enabled device may transfer information or make payments to another NFC compatible device by coming near within a few centimeters of one another without physically contacting each other. This type of technology is conventionally referred to as "contactless" technology and a payment made with this technology is referred to as "contactless" payment.

However, regardless of benefits that may be obtained through integrating wallet functionality into mobile device, prevailing technology still lacks an effective means to manage various payment applets residing within the mobile device.

2

With the advent of NFC-based contactless payment applications, users were provided a way to select a contactless payment applet (i.e., contactless payment virtual card) from various contactless payment applets stored in the mobile device for payment at corresponding point-of-sale (POS) devices. However, while these contactless payment applets may be selected to make a purchase, the management of payment applets may be limited. For example, a user may be limited to view the contactless payment applets stored in the user's mobile device when interacting with a POS device. Further, even if the user is able to view the various contactless payment applets stored in the mobile device with or without the POS device, the user may be unable to view the details related to the contactless payment applets (e.g., account number, expiration date, security code, balance and the like). Accordingly, users may be unable to effectively manage or keep track of various contactless payment applets stored in their respective mobile devices.

Typically, the contactless card applets may be stored within a specific compartment, or a secured domain, of the SE to be accessed during an interaction with the POS device. Moreover, even when such payment applications are accessed, since these applications are managed through industry standard Payment Procedure Secure Elements (PPSE) that only provide for application identification (ID) and label, a limited generic description may be provided to the user. Accordingly, the user may be unable to view any account specific information stored within the SE or manage such applications with or without the use of POS equipment.

Another limitation of current mobile wallet applications is the lack of support providing for such technology. With such focus on mobile commerce, many competing service providers seek delivering their services to the users. However, such services may be offered to the users without regard to the mobile device capabilities or mobile service providers utilized by the user. Due to technical or business compatibility, there may be numerous applications that may be inapplicable to the user's individual attributes (e.g., bank membership, mobile service provider, manufacturer of a mobile device owned by the user, type of secure element installed in the mobile device, operating system of the mobile device, and the like). Accordingly, users may often be bombarded with various applications that may be inapplicable to the user, making the process more difficult than necessary.

Another issue with the current mobile wallet application is its ability to update its information. As various service providers operate independently from one another, when an update is required by a particular service provider, each individual application is typically updated separately. Such inefficiency may dissuade users from obtaining crucial updates that may be necessary to a particular application.

SUMMARY

Exemplary embodiments of the present invention provide a mobile device to store a mobile wallet application and a wallet management system (WMS) to store corresponding wallet application information. Exemplary embodiments of the present invention provide a method for provisioning a wallet application, a contactless card applet, a wallet management applet (WMA), and a widget. Exemplary embodiments of the present invention provide a method for synchronizing a mobile wallet application with the WMS.

Additional features of the invention will be set forth in the description which follows, and in part will be apparent from the description, or may be learned by practice of the invention.

US 8,843,125 B2

3

Exemplary embodiments of the present invention provide a method for installing a wallet application in a mobile device including requesting, by the mobile device, a mobile wallet application comprising a corresponding Over-the-Air (OTA) proxy; receiving mobile wallet application installation information; installing the mobile wallet application in the mobile device; capturing mobile device information by using the OTA proxy, the mobile device information comprising secure element (SE) information; and transmitting the mobile device information for registering the installed mobile wallet application.

Exemplary embodiments of the present invention provide a method for managing mobile wallet accounts installed on a mobile devices including receiving a request for a mobile wallet application from a mobile device; transmitting the mobile wallet application to the mobile device; receiving mobile device information, the mobile device information comprising SE information; and registering the mobile device and the corresponding mobile wallet application in a trusted service manager (TSM).

Exemplary embodiments of the present invention provide method for provisioning a contactless card applet in a mobile device comprising a mobile wallet application including activating the mobile wallet application; connecting to a TSM system; synchronizing the mobile wallet application with the TSM system; displaying a contactless card applet based on attributes of the mobile device; receiving a selection of a contactless card applet; retrieving a widget and a WMA corresponding to the contactless card applet; and provisioning the selected contactless card applet, widget, and the WMA.

Exemplary embodiments of the present invention provide a WMS in a non-transitory storage medium to store and manage mobile wallet account information including a wallet client management component to store and to manage a mobile wallet application; a widget management component to store and to manage widgets; a device profile management component to store mobile device information; and a rule engine to filter a widget based on the mobile device information.

Exemplary embodiments of the present invention provide a mobile device including a SE; a mobile wallet application to store a widget corresponding to a contactless card applet, wherein the contactless card applet is stored in the SE; a WMA corresponding to the contactless card applet, wherein WMA is stored in the SE; and an OTA proxy to provision the contactless card applet, a widget corresponding to the contactless card applet, and the WMA.

It is to be understood that both foregoing general descriptions and the following detailed description are exemplary and explanatory and are intended to provide further explanation of the invention as claimed. Other features and aspects will be apparent from the following detailed description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are included to provide a further understanding of the invention and are incorporated in and constitute a part of this specification, illustrate embodiments of the invention, and together with the description serve to explain the principles of the invention.

FIG. 1 is a system diagram of a mobile wallet application and associated integration in accordance with an exemplary embodiment of the present invention.

FIG. 2 is a system diagram illustrating a system and method for provisioning mobile card wallet management application along with supporting applications, mobile card

4

widgets, contactless card applets, and related credentials in accordance with an exemplary embodiment of the present invention.

FIG. 3 is a system diagram illustrating a system and method for provisioning service provider specific mobile card widgets, contactless card applets, and wallet management application account information in accordance with an exemplary embodiment of the present invention.

FIG. 4 is a system diagram illustrating a system and method for dynamically filtering applicable mobile wallet service provider specific widgets based upon user account attributes in accordance with an exemplary embodiment of the present invention.

FIG. 5 is a system diagram illustrating a system and method for synchronizing mobile wallet application with the master mobile wallet configuration server to provide a most current version of the mobile wallet application in accordance with an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE ILLUSTRATED EMBODIMENTS

The invention is described more fully hereinafter with references to the accompanying drawings, in which exemplary embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather, these exemplary embodiments are provided so that this disclosure is thorough, and will fully convey the scope of the invention to those skilled in the art. It will be understood that for the purposes of this disclosure, "at least one of each" will be interpreted to mean any combination of the enumerated elements following the respective language, including combination of multiples of the enumerated elements. For example, "at least one of X, Y, and Z" will be construed to mean X only, Y only, Z only, or any combination of two or more items X, Y, and Z (e.g. XYZ, XZ, YZ). Throughout the drawings and the detailed description, unless otherwise described, the same drawing reference numerals are understood to refer to the same elements, features, and structures. The relative size and depiction of these elements may be exaggerated for clarity, illustration, and convenience.

FIG. 1 is a system diagram of a mobile wallet system and associated integration, according to an exemplary embodiment of the present invention.

As shown in FIG. 1, an example system utilizing mobile wallet technology may include a mobile device **100**, mobile wallet management system (WMS) **110**, supporting Trusted Service Manager (TSM) system **120**, Mobile Network Operator (MNO) **130**, and Service Provider (SP) **140**.

WMS **110** includes a wallet client management component **111**, widget management component **112**, device profile management component **113**, user profile management component **114**, data management component **115**, and rule engine **116**.

In particular, wallet client management component **111** is responsible for the wallet application itself (referred as the container), which may house the individual widgets (e.g., applications stored at the application level related to a financial institution, transportation account, and the like). The wallet client management component **111** may store container specific information, including the type of wallet application and manufacturer. For example, wallet client management component **111** may recognize a user John has a mobile wallet application manufactured by Google® and has specified set of known functionalities. By managing the type of

US 8,843,125 B2

5

wallet application the user has on the mobile device, it may be possible to provide the same wallet application when necessary.

Widget management component 112 on the other hand is responsible for the individual widgets stored within the wallet container. Widgets may be an application configured to interface with a user of the mobile device. In an example, widgets may refer to individual payment applications, transportation applications, and other related applications. Device Profile management component 113 houses a memory to store one or more programs, such as applications, and other related information. Device Profile management component 113 may store device specific information, such as information related to the mobile device itself including type of mobile device, supporting operating system (OS), mobile service provider, and other relevant information. User Profile management component 114 captures user identifying information such as name, address, birthday, phone number, and the like. Data Management component 115 allows further expansion of data management services offered by a mobile WMS (e.g., transaction history, user preferences, loyalty programs, digital receipts, digital coupons and the like). Rule engine 116 may filter widgets based on information related to the mobile device. Although various components were illustrated to be included in the WMS 110, the configuration of WMS 110 is not limited thereto. The illustrated components may be included within the WMS 110 or external to the WMS 110.

The disclosed WMS 110 may reside within TSM system 120 or independent of the TSM system 120. For the purposes of this disclosure, it will be assumed that the WMS 110 is housed within the TSM system 120. Like the TSM system 120, WMS 110 may interact with MNO 130 to transmit and receive billing related information. Further, WMS 110 may interact with SP 140 to receive and transmit SP payment card information.

The TSM system 120 may refer to a third party entity positioned to consolidate various information from various service providers including, financial institutions, MNOs, handset manufacturers, and card manufacturers. As TSM system 120 may hold various information from various parties, the mobile device may interact with the TSM system individually rather than various discrete entities. Accordingly, the described TSM system 120 may act as an integration point for all of the external parties the mobile device may deal with, providing for a seamless and more efficient operation of mobile services.

A method for installing a mobile wallet application and associated management applet in a secure element (SE) is described below in reference to FIG. 2. FIG. 2 is a system diagram illustrating a system and method for installing a mobile wallet application on the mobile device and correlating wallet management applet in the SE of the mobile device in accordance with an exemplary embodiment of the present invention.

As shown in FIG. 2, in step 201, the mobile device 100 requests a new mobile wallet application 24. In an alternative flow, a SP 140 may request installation of the mobile wallet application 24 from the TSM system 120. When requesting installation of mobile wallet application 24 from the TSM system 120, the TSM system 120 may wait for the mobile device 100 to connect to the TSM system 120 before installing the mobile wallet application 24. The TSM system 120 may install the mobile wallet application 24 directly upon connection to the mobile device 100 or wait until the user approves the request to install the mobile wallet application 24. If installation is executed, a corresponding widget representing a virtual card, such as a virtual credit card, may be

6

provisioned to reside within the respective mobile wallet application 24. In an example, the widget representing the virtual card may reside within the mobile wallet application 24.

Once request is made, in step 202, the TSM system 120 receives the mobile wallet application installation request and corresponding identification information and checks for duplicate records existing in the TSM system 120. If it is determined that the requesting customer is a new customer, a new record is created within the TSM system 120. If the customer information already exists, TSM system 120 may verify the existing customer and update the customer's information, if applicable.

After a customer account has been created or updated, if it is determined that the mobile wallet application 24 is not installed on the mobile device 100, the TSM system 120 will confirm the mobile wallet application installation request and initiate the wallet application installation process. The installation process may be initiated by transmitting a Wireless Application Protocol (WAP) message with an embedded Uniform Resource Locator (URL) to the Short Message Service (SMS) platform in step 203, which relays the message to the mobile device 100 in step 204. However, the mobile wallet application 24 may be obtained in various other ways as well and is not limited to the WAP message method as described above. The mobile wallet application 24 may be downloaded directly to the requesting mobile device 100, sent to the user in a physical medium storing the application, or by other suitable methods for providing software applications.

The user, upon receipt of the installation message from the SMS platform, may initiate the actual installation process by sending a request to the TSM system 120 in step 205.

In response, TSM system 120 transmits the requested mobile wallet application 24 to mobile device 100 for installation and an accompanying over-the-air (OTA) proxy program to allow OTA provisioning in step 206. Although mobile wallet application 24 and OTA proxy are shown as being part of mobile device 100, an ordinarily skilled artisan understands that these elements may not be present on mobile device 100 until they are installed.

Once the mobile wallet application 24 and accompanying OTA proxy program have been downloaded, the mobile wallet application 24 may be launched by the requesting user in step 207. Alternatively, the mobile wallet application 24 may be launched automatically once it is downloaded. Also, in the event OTA proxy is already downloaded or installed, the mobile wallet application 24 may be downloaded independently of the OTA proxy. Although not illustrated, the accompanying OTA proxy may be included in the mobile wallet application 24.

In step 208, the OTA proxy captures the mobile device information (e.g. International Mobile Equipment Identity (IMEI)/Mobile Equipment Identifier (MEID), Mobile Subscriber Integrated Services Digital Network Number (MSISDN)), including SE information (e.g. Card Production Life Cycle (CPLC), Card Serial Number (CSN), Card Image Number (CIN), Integrated Circuit Card Identification (ICCID)), which may be stored in a device memory component of the mobile device 100. The OTA proxy may be a separate component from the mobile wallet application 24, or may be included in the mobile wallet application 24.

Afterwards, in step 209, the OTA proxy sends the captured SE and mobile device information to the TSM system 120, which may house a WMS 110 (as shown in FIG. 2) or be in communication with an external WMS 110 (as shown in FIG. 1).

US 8,843,125 B2

7

The WMS 110, upon receipt of the information provided by the OTA proxy, creates a Mobile identification (ID) for the installed mobile wallet application 24 in step 210. Once the mobile ID has been created, the WMS 110 requests TSM system 120 to provision a corresponding wallet management applet (WMA) 21 with the following information via OTA proxy: CPLC or CSN, CIN, Mobile ID and WMA personalization data. In an example, WMA 21 may include both a WMA 21 container and one or more WMA 21 applets. WMA 21 container may manage the information stored in the WMA 21 applets. WMA 21 container may be installed in the mobile device 100 when WMA 21 applet is requested to be installed, or when the mobile wallet application is installed, or separately without regard to either the WMA 21 applet or the mobile wallet application.

The WMA 21 container is a software application that may reside within the SE of the mobile device 100 to manage account information related to the contactless card applet 23 (i.e. WMA 21 applet) that may be typically inaccessible by the user. In an example, the SE may store one or more contactless card applets that may be used through a mobile device 100 with NFC capability, but the contactless card applets may largely be inaccessible by the user. More specifically, during a financial transaction, the NFC enabled mobile device may transmit contactless card information, which may include account specific information to a POS device to complete the transaction. However, even during this transaction, the user is typically limited to the selection of a generic logo corresponding to the contactless card applet being used in the transaction, but no account specific information may be accessed by the user of the mobile device 100. In an example, account specific information may include credit card number, expiration date, security code (e.g., a combination of numbers typically found on back of credit cards), personal identification number (PIN) (e.g., a combination of numbers typically used to conduct financial transactions with the user's financial institution), and other related information.

To provide the user of the mobile device with the account specific information related to contactless card applets, separate account information associated with the corresponding contactless card applet 23 (e.g. credit card number, expiration date, security code, PIN, etc.) may be provisioned into the SE as WMA 21 applets. The respective account information or WMA 21 applet may be provided by duplicating the account information associated with the contactless card when the TSM system receives contactless card applets from SPs to provision into the mobile device 100. Alternatively, SP providing the contactless card applet may provide the account related information separately to the TSM system for provisioning.

In step 211, TSM system 120 sends a wake up message to the mobile push server (e.g. Cloud to Device Messaging (C2DM)) with a mobile device identifier to wake up OTA proxy residing in the requesting mobile device 100.

The mobile push server routes the received message to the mobile wallet application 24, which in turn sends the request to OTA proxy and wakes OTA proxy in step 212.

In step 213, the OTA proxy gathers mobile device and SE specific information such as MSISDN and CIN and sends it over to TSM system 120. In an example, OTA proxy gathers mobile device and SE specific information to send to TSM system 120 every time it is woken up. Alternatively, this step may be skipped and the mobile device and SE information provided in step 209 to register the mobile device 100 and the wallet application may be used.

Once TSM system 120 receives the information sent by OTA Proxy in step 213, TSM system 120 processes the infor-

8

mation and converts the identifying information along with the request to provision WMA 21 container into Application Protocol Data Unit (APDU) commands in step 214 and sends them over to OTA proxy in step 215.

Next, in step 216, OTA proxy receives the APDU commands to install WMA 21 container and relays them to the SE, which processes the APDU commands to install the requested WMA 21 container and its associated credentials. SE then responds back with the result of each command request in step 217. Although WMA 21 container, PPSE 22, and Contactless Card Applet 23 are shown as being part of mobile device 100, an ordinarily skilled artisan understands that these elements may not be present on the SE of the mobile device 100 until they are installed.

Subsequently, OTA Proxy relays the result back to the TSM system 120 in step 218, and the TSM system 120 updates its system with the result.

Once the mobile wallet application 24 has been successfully installed in the mobile device 100, the user may provision SP 140 specific contactless card applets 23, and its corresponding widget applications and WMA 21 applet onto mobile device 100.

FIG. 3 is a system diagram illustrating a method for installing a mobile widget into the mobile wallet application 24 and its corresponding contactless card applet and account information into the SE of the requesting mobile device in accordance with an exemplary embodiment of the present invention.

In step 301, the user logs into the mobile wallet application 24 to start the mobile wallet application 24 for use. Once started, the mobile wallet application 24 connects to the TSM system 120, which may house WMS 110, for synchronization in step 302. A more detailed description of how this synchronization process operates is provided below with reference to FIG. 5.

TSM system 120 checks for any updated information made by external parties (e.g. SP 140, user by web access, TSM system 120 administrator, and/or etc.) and sends the list of waiting updates to the mobile wallet application 24 in step 303. Further, additional applications that user may be interested in may be displayed for download through dynamic filtering. The applicable applications based on user attributes will be displayed through this filtering process. A more detailed description of how this dynamic filtering works is provided below with reference to FIG. 4.

The mobile device user is prompted to decide whether to update the mobile wallet application 24 with the changes made at the TSM system 120, if any, in step 304. Alternatively, the mobile device may update the mobile wallet application 24 automatically with the respective changes in step 304.

When the mobile device 100 updates the mobile wallet application 24 or downloads a new application, a request is made to the TSM system 120/WMS 110 to provision the updates and/or selected card applications in step 305. If a request to update requires updating of account specific information, such as change in account number or expiration date, the process to update the application will follow the same steps regardless of the information being updated.

Further, if a request to provision the selected contactless card applet 23 is made, such as a "VISA®" contactless card applet, a corresponding widget and WMA 21 applet may be programmed to be provisioned automatically. The corresponding widget may reside in the mobile wallet application 24, at the application level, to provide an interface to the user. The corresponding WMA 21 applet, which may include account specific information of the contactless card applet

US 8,843,125 B2

9

(e.g. credit card number, expiration date, security code, PIN, etc.), may be provisioned into the SE. By installing both the WMA 21 applet and the widget, the user may view and manage the information stored in the WMA 21 applet through the corresponding widget.

TSM system 120 processes the provisioning request and sends a wake up message request to the mobile push server in step 306, and the push server proceeds to relay the request the mobile wallet application 24, which in turn sends the message to OTA proxy, thereby waking OTA proxy in step 307.

In step 308, OTA proxy wakes up and gathers mobile device and SE specific information, such as MSISDN and CIN, and sends the collected information to TSM system 120.

Once TSM system 120 receives the information sent by OTA Proxy, TSM system 120 processes the received information along with the provisioning command and converts both the received information along with the provisioning command into APDU commands to send to OTA proxy in step 309. When sending the APDU commands, the contactless card applet and the corresponding WMA 21 applet are sent to OTA proxy for provisioning into the SE. However, since the widget is provisioned at the application level and not into the SE, the widget may be provisioned through the OTA proxy or through a wireless network.

Next, in step 310, OTA proxy receives the APDU commands from the TSM system 120 to install requested issuer contactless applets 23 and correlating WMA 21 applet to be provisioned. In an example, contactless applets 23 and correlating WMA 21 applet are provided in different domains of the same SE. In response, SE processes the APDU commands for both the contactless applet and the WMA 21 applet and sends back the result of each command request in step 311. As APDU commands may be processed one at a time, multiple communications may be passed back and forth between the OTA proxy and the SE.

Subsequently, OTA Proxy relays the result back to the TSM system 120 in step 312, and the TSM system 120 updates its system with the result of the request in step 313. Once information is updated, notification of the results is sent to SP 140 in step 314. Similarly, the mobile wallet application 24 notifies TSM system 120 of the result of the widget installation. For example, the mobile wallet application 24 will notify the TSM system 120 whether the widget installation was a success or a failure.

Once account specific information is installed into WMA 21 container as WMA 21 applet, the respective mobile device 100 may access the information periodically for required updates. For example, the mobile device 100 may access the information stored in the WMA 21 applet using the mobile wallet application 24 to check for the expiration dates of the contactless card applets 23 stored within the mobile device 100 and prompt user for updates as necessary. Alternatively, the mobile wallet application 24 may check for updates automatically. In addition, the user may also gain access to the account number, security code, and corresponding expiration date as necessary to make purchases online for further utility. In an example, the information stored in the WMA 21 applet may allow the mobile device 100 to check the expiration date of the contactless card applet 23 and request update when the card applet expires.

WMA 21 container may, however, limit amount of change requests to the WMA 21 applet as they contain account specific information. For example, the number of times expiration date may be changed with a reference time period may be limited, or changes to the credit card numbers may be prohibited. In addition, WMA 21 container may prevent user from making changes directly in the WMA 21 applet but

10

allow request for modification to the TSM system 120, which in turn will make the request to the relevant external parties. While the described process illustrates a preferred embodiment of the present invention, the amount of modification allowed by the WMA 21 container is not limited to what has been described. In some instances, WMA 21 container may allow direct modification to the account specific information as dictated by business needs.

FIG. 4 is a system diagram illustrating a method for dynamically filtering a list of mobile widget applications that are available for installation based upon corresponding mobile device attributes in accordance with an exemplary embodiment of the present invention.

In step 401, the user logs into the mobile wallet application 24, which seeks to connect with the TSM system 120/WMS 110.

The TSM system 120 receives the connection request through a mobile gateway residing within the TSM system 120 and relays the request to a Rule Engine in TSM system 120 in step 402. The TSM system 120 queries the user account in its system in step 403 for equipment information, MNO, SP accounts, and any other relevant information. Based on the mobile device 100 attributes, a filtered list of downloadable applications from the TSM system 120 may be displayed to the mobile device. In an example, mobile device 100 attributes may include, without limitation, the mobile network provider of the mobile device 100 (e.g. "Sprint®," "Verizon®", "AT&T®", etc.), financial institutions associated with the contactless card applets stored (e.g. "Wachovia®," "Bank of America®," "Chase®", etc.), mobile device 100 manufacturer (e.g. "HTC®", "Motorola®", "Apple®", etc.), and mobile device 100 hardware specifications (i.e. hardware, software, operating system, etc.).

Here, TSM system 120 may house a large list of available applications, including contactless card applets 23, as illustrated in FIG. 4. TSM system 120 may house various applications without regard to the device capabilities, SPs' relationship with other SPs, or other limitations that may be inherent in the business or technical environments. However, as an individual user connects with the TSM system 120 to download new applications, TSM system 120 may dynamically filter the list of available applications based upon the mobile device attributes described above.

As many mobile devices operate with various operating systems and standards, not all of the applets provided by the SP may be compatible with the user mobile device or user's MNO. Because of lack of standardization of hardware and software on mobile devices, an efficient method to filter only the relevant applets is helpful. Along with these technical limitations, many MNOs and SPs may decide not to provide their services to each other for business reasons. As the general public may not be familiar with such knowledge, an additional filtering mechanism may be provided to provide only the applicable applets to the requesting user. In an example, all of the provided limitations may be managed and applied by the Rule Engine in the TSM system 120. The Rule Engine may be housed in the TSM system 120 or may exist as an external entity, which interacts with TSM system 120 through a network. Further, the Rule Engine may be a combination of software and hardware, software to apply and manage the rules and hardware to store the relevant rules. Accordingly, by providing an active dynamic filtering mechanism at the TSM system 120 level, all of the parties involved in such transaction need to make only a general request to the TSM system 120 to access and to provide customer specific services.

US 8,843,125 B2

11

Once the list of applicable applets have been dynamically filtered, TSM system 120 sends the list of applets to display to the mobile gateway in step 404, which relays it back to the mobile wallet application 24 in step 405.

In FIG. 5, a system diagram is provided for synchronizing the mobile wallet application residing within the mobile device with the TSM system in accordance with an exemplary embodiment of the present invention. As with many electronic devices that may be prone to damage and wear, or often misplaced, a centralized management or storage may be beneficial to maintain a master file of the user wallet configuration.

In step 501, multiple external parties, such as credit card service providers as illustrated in FIG. 5, may send a request for changes to be made to the user's mobile wallet configuration directly to the TSM system 120/WMS 110, which may store the master configuration of the respective mobile wallet application 24. In addition, TSM system 120 administrators and the user themselves may access the TSM system 120 via web access or any other remote access functionality. As the mobile wallet application 24 may not always be on, a central repository allows external parties to make the necessary requests without regard to user's mobile wallet application 24's operating status. For example, SPs 140 may request an additional contactless card applet 23 to be provisioned to the user's SE on their own time without regard to the mobile wallet application 24's operating status. Similarly, TSM system 120 itself may recognize that the expiration date of the respective application is coming up and prompt the user to update the card for provisioning when the mobile wallet application 24 connects to the system.

While only TSM system 120 administrator, SP 140, and the user were displayed, the requesting party may be any external party to the TSM system 120.

Subsequently, in step 502, when the user logs into the mobile wallet application 24, the mobile wallet application 24 checks with the TSM system 120/WMS 110 for any modifications to the wallet configuration since the last login by the user. As the mobile wallet application 24 synchronizes every time the application is logged into, the user can be sure that the user has access to the most current information during use. In addition, by limiting synchronization events to access of mobile wallet application 24, secure access to sensitive information is provided only when the user is utilizing the mobile wallet application 24. However, if desired, mobile wallet application 24 may always be in sync by automatically whenever mobile device is on and has mobile signal.

Any updates made in the WMS 110 while mobile wallet application 24 was offline will be prompted for the user to make the updates in step 503. User may update one application at a time or all at once if such is desired. Also, the user may set the application to automatically update every change made in the TSM system 120/WMS 110 at synchronization.

In step 504, while mobile wallet application 24 is still active, any modifications that are made in the mobile wallet application 24 itself will be updated in the WMS 110 in step 505 as synchronization is a continuous one during usage. For example, if the user changes a user preference on the mobile wallet application 24, changes to the user preference may be updated into the WMS 110 in real time. Similarly, if the mobile device 110 prompts the user to update the expiration date of the contactless applet and the user agrees, user's request will be submitted to TSM system 120, which will process the request and route it to SP 140 for processing.

It will be apparent to those skilled in the art that various modifications and variation can be made in the present invention without departing from the spirit or scope of the inven-

12

tion. Thus, it is intended that the present invention cover the modifications and variations of this invention provided they come within the scope of the appended claims and their equivalents.

What is claimed is:

1. A method for installing a wallet application in a mobile device, comprising:

requesting, by the mobile device, a mobile wallet application comprising a corresponding Over-the-Air (OTA) proxy;

receiving mobile wallet application installation information;

installing the mobile wallet application in the mobile device;

capturing mobile device information by using the OTA proxy, the mobile device information comprising secure element (SE) information; and

transmitting the mobile device information for registering the installed mobile wallet application.

2. The method of claim 1, wherein installing the mobile wallet application in the mobile device comprises automatically installing upon receipt of the mobile wallet application installation information.

3. The method of claim 1, further comprising provisioning a wallet management applet (WMA) container into the SE of the mobile device.

4. The method of claim 1, wherein receiving mobile wallet application installation information comprises:

receiving a Wireless Application Protocol (WAP) message with an embedded Uniform Resource Locator (URL) from a Short Message Service (SMS) platform.

5. The method of claim 1, wherein capturing mobile device information comprises:

capturing at least one of an International Mobile Equipment Identity (IMEI), a Mobile Equipment Identifier (MEID), a Mobile Subscriber Integrated Services Digital Network Number (MSISDN), a Card Production Life Cycle (CPLC), a Card Serial Number (CSN), a Card Image Number (CIN), and an Integrated Circuit Card Identification (ICCID).

6. The method of claim 3, wherein provisioning a WMA container into the SE comprises:

transmitting a request to provision the WMA container; and

receiving the WMA container installation information in Application Protocol Data Unit (APDU) commands; and

provisioning the converted APDU commands to the SE.

7. A method for managing mobile wallet accounts installed on mobile devices, comprising:

receiving a request for a mobile wallet application from a mobile device;

transmitting the mobile wallet application to the mobile device;

receiving mobile device information, the mobile device information comprising secure element (SE) information; and

registering the mobile device and the corresponding mobile wallet application in a trusted service manager (TSM).

8. The method of claim 7, wherein registering the mobile device and the corresponding mobile wallet application in a TSM comprises:

checking for registered account information corresponding to the requesting mobile device in the TSM system; and registering the mobile device in the TSM system in response to no corresponding registered account infor-

US 8,843,125 B2

13

mation being found in the TSM system, or updating account information in response to finding the corresponding registered account information in the TSM system.

9. The method of claim 7, further comprising transmitting an accompanying over-the-air (OTA) proxy application to the mobile device.

10. The method of claim 8, wherein registering the mobile device comprises:

creating a mobile identifier for the installed mobile wallet application;
storing the mobile device information; and
connecting the mobile device information with the mobile identifier.

11. A method for provisioning a contactless card applet in a mobile device comprising a mobile wallet application, the method comprising:

activating the mobile wallet application;
connecting to a Trusted Service Manager (TSM) system;
synchronizing the mobile wallet application with the TSM system;
displaying a contactless card applet based on attributes of the mobile device;
receiving a selection of a contactless card applet;
retrieving a widget and a wallet management applet (WMA) corresponding to the contactless card applet; and
provisioning the selected contactless card applet, the widget, and the WMA.

12. The method of claim 11, wherein synchronizing the mobile wallet application with the TSM system comprises: receiving a change made to a mobile wallet application user account on the TSM system; and provisioning the changed information.

13. The method of claim 11, wherein synchronizing the mobile wallet application with the TSM system comprises: checking for a change made to a configuration of the mobile wallet application; and transmitting the change to the TSM system.

14. The method of claim 11, wherein displaying a contactless card applet based on attributes of the mobile device comprises:

retrieving mobile device information comprising SE information;
transmitting the mobile device information; and
receiving filtered contactless card applet for provisioning, wherein the contactless card applet is filtered based on the mobile device information.

15. The method of claim 14, wherein displaying the contactless card applet further comprises:

receiving filtered contactless card applet for provisioning, wherein the contactless card applet is filtered based on the business rules.

16. The method of claim 11, wherein provisioning the selected contactless card applet, WMA and widget comprises:

transmitting a request for installation of the contactless applet and the corresponding widget and WMA to be installed, wherein the WMA is a software application configured to store account specific information and the widget is an application configured to interface with a user of the mobile device; and

14

receiving the contactless applet, the WMA, and the widget information through OTA proxy.

17. The method of claim 16, wherein account specific information comprises at least one of a payment card number, a security code, an expiration date, and a personal identification number (PIN).

18. A wallet management system (WMS) in a non-transitory storage medium to store and manage mobile wallet account information, comprising:

a wallet client management component configured to store and to manage a mobile wallet application;
a widget management component configured to store and to manage widgets;
a device profile management component configured to store mobile device information; and
a rule engine configured to filter a widget based on the mobile device information,
wherein said wallet management system is configured to receive the mobile device information from a mobile device and store the mobile device information in the device profile management component, and
wherein said wallet management system is configured to register the mobile device and the mobile wallet application in a Trusted Service Manager (TSM) system.

19. The WMS of claim 18, wherein the wallet client management component further stores wallet application specific information comprising at least wallet application type and wallet application manufacturer information.

20. The WMS of claim 18, wherein the mobile device information comprises at least one of a mobile device type, a supporting Operating System (OS), a mobile service provider, a mobile device manufacturer, and a secure element (SE) type.

21. The WMS of claim 18, further comprising a user profile management component to capture and manage user identifying information.

22. The WMS of claim 18, wherein the WMS is hosted on the TSM system.

23. A mobile device, comprising:
a secure element (SE);

a mobile wallet application configured to store a widget corresponding to a contactless card applet, wherein the contactless card applet is stored in the SE;

a wallet management applet (WMA) corresponding to the contactless card applet, wherein the WMA is stored in the SE; and

an over-the-air (OTA) proxy configured to provision the contactless card applet, a widget corresponding to the contactless card applet, and the WMA,

wherein said OTA proxy is configured to capture mobile device information comprising SE information; and

wherein said OTA proxy is configured to transmit the mobile device information for registering the mobile wallet application.

24. The mobile device of claim 23, wherein WMA is configured to store account information associated with the contactless card applet, and the widget is configured to include a user interface.

25. The mobile device of claim 24, wherein the account information comprises at least one of a card number to access financial information, a security code, a personal identification number (PIN), and an expiration date.

* * * * *

FORM 31. Certificate of Confidential Material

Form 31
July 2020

**UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT**

CERTIFICATE OF CONFIDENTIAL MATERIAL

Case Number: 23-2208

Short Case Caption: FINTIV, INC. v. APPLE, INC.

Instructions: When computing a confidential word count, Fed. Cir. R. 25.1(d)(1)(C) applies the following exclusions:

- Only count each unique word or number once (repeated uses of the same word do not count more than once).
- For a responsive filing, do not count words marked confidential for the first time in the preceding filing.

The limitations of Fed. Cir. R. 25.1(d)(1) do not apply to appendices; attachments; exhibits; and addenda. *See* Fed. Cir. R. 25.1(d)(1)(D).

The foregoing document contains 15 number of unique words (including numbers) marked confidential.

- ☒ This number does not exceed the maximum of 15 words permitted by Fed. Cir. R. 25.1(d)(1)(A).
- ☐ This number does not exceed the maximum of 50 words permitted by Fed. Cir. R. 25.1(d)(1)(B) for cases under 19 U.S.C. § 1516a or 28 U.S.C. § 1491(b).
- ☐ This number exceeds the maximum permitted by Federal Circuit Rule 25.1(d)(1), and the filing is accompanied by a motion to waive the confidentiality requirements.

Date: 11/16/2023

Signature: /s/ Meredith Martin Addy

Name: Meredith Martin Addy

CERTIFICATE OF SERVICE

I hereby certify that on November 16, 2023, I filed and served Appellant Fintiv, Inc.'s Non-Confidential Opening Brief via the CM/ECF system.

/s/ Meredith Martin Addy

Meredith Martin Addy